IS 10767-1 (2001): Programmable Measuring Instruments -- Interface System (Byte Serial, Bit Parallel), Part 1: Functional, Electrical and Mechanical Specifications, System Applications and Requirements for the Designer and User [LITD 8: Electronic Measuring Instruments, Systems and Accessories]

BLANK PAGE

IS 10767 (Part 1) : 2001
IEC 60625-1 (1993)

भारतीय मानक

प्रोग्रामेबल मापन उपकरण — इंटरफेस तंत्र (बाइट सीरियल, बिट पैरलल)

भाग 1 अभिकल्पक तथा प्रयोक्ता के लिए कार्यात्मक, विद्युतीय एवं यांत्रिकी विशिष्टियाँ, तंत्र अनुप्रयोग एवं अपेक्षाएँ

(पहला पुनरीक्षण)

Indian Standard

PROGRAMMABLE MEASURING INSTRUMENTS — INTERFACE SYSTEM (BYTE SERIAL, BIT PARALLEL)

PART 1 FUNCTIONAL, ELECTRICAL AND MECHANICAL SPECIFICATIONS, SYSTEM APPLICATIONS AND REQUIREMENTS FOR THE DESIGNER AND USER

(First Revision)

ICS 35.200; 35.240.50; 25.040.40

BUREAU OF INDIAN STANDARDS
MANAK BHAVAN, 9 BAHADUR SHAH ZAFAR MARG
NEW DELHI 110002

December 2001 **Price Rs 700.00**

Electronic Measuring Instruments, Systems and Accessories Sectional Committee, LTD 21

NATIONAL FOREWORD

This Indian Standard (Part 1) (First Revision) which is identical with IEC 60625-1 (1993) 'Programmable measuring instruments — Interface system (byte serial, bit parallel) — Part 1 : Functional, electrical and mechanical specifications, system applications and requirements for the designer and user' issued by the International Electrotechnical Commission (IEC) was adopted by the Bureau of Indian Standards on the recommendation of Electronic Measuring Instruments, Systems and Accessories Sectional Committee and approval of the Electronics and Telecommunication Division Council.

This standard was first published in 1984 and was identical to IEC 60625-1 (1979) 'An interface system for programmable measuring instruments (byte serial, bit parallel) — Part 1 : Functional, electrical and mechanical specifications, system applications and requirements'. IEC 60625-1 (1979) has since been revised in 1993. In view of the technological advances at the international level, this standard is being revised to align it with the latest international practices.

The text of the IEC has been approved as suitable for publication as Indian Standard without deviations. Certain conventions are, however, not identical to those used in Indian Standards. Attention is particularly drawn to the following:

a) Wherever the words 'International Standard' appear referring to this standard, they should be read as 'Indian Standard'.

b) Comma (,) has been used as a decimal marker while in Indian Standards, the current practice is to use a point (.) as the decimal marker.

CROSS REFERENCES

In the adopted standard, reference appears to certain International Standards for which Indian Standards also exist. The corresponding Indian Standards which are to be substituted in their place are listed below along with their degree of equivalence for the editions indicated:

International Standard	*Corresponding Indian Standard*	*Degree of Equivalence*
IEC 60050-351 (1975) International Electrotechnical Vocabulary — Chapter 351 : Automatic control	IS 1885 (Part 75) : 1993 Electrotechnical vocabulary : Part 75 Automatic control	Identical
IEC 60068-1 (1988) Environmental testing — Part 1 : General and guidance	a) IS 9000 (Part 1) : 1988 Basic environmental testing procedures for electrical and electronics items : Part 1 General b) IS 9001 (Part 1) : 1984 Guidance for environmental testing : Part 1 General	Technically equivalent
IEC 60359 (1987) Expression of the performance of electrical and electronic measuring equipment	IS 9176 : 2001 Expression of the performance of electrical and electronic measuring equipment	Identical

International Standard	*Corresponding Indian Standard*	*Degree of Equivalence*
IEC 60512-6 (1984) Electromechanical components for electronic equipment, basic testing procedures and measuring methods — Part 6 : Climatic tests and soldering tests	IS 12248 (Part 6) : 1991 Basic testing procedures and measuring methods of electromechanical components for electronic equipment : Part 6 Climatic tests and soldering tests	Technically equivalent
IEC 60625-2 (1993) Programmable measuring instruments — Interface system (byte serial, bit parallel) — Part 2 : Codes, formats, protocols and common commands	IS 10767 (Part 2) : 2001 Programmable measuring instruments — Interface system (byte serial, bit parallel) : Part 2 Codes, formats, protocols and common commands	Identical
ISO/IEC 646 (1991) Information technology — ISO 7-bit coded character set for information interchange	IS 10315 : 1997 ISO 7-bit coded character set for information interchange	do

The concerned Technical Committee responsible for preparation of this standard has reviewed the provisions of the following International publications and has decided that they are acceptable for use in conjunction with this standard:

IEC 60348 (1978)	Safety requirements for electronic measuring apparatus [Since withdrawn and replaced by IEC 61010-1 (1990)]
IEC 60603–2 (1988)	Connectors for frequencies below 3 MHz for use with printed boards — Part 2 : Two-port connectors for printed boards, for basic grid of 2.54 mm (0.1 in) with common mounting features
IEC 60807 – 2 (1992)	Rectangular connectors for frequencies below 3 MHz — Part 2 : Detail specification for a range of connectors, with assessed quality with trapezoidal shaped metal shells and round contacts — fixed solder contact types

Only the English language text of the International Standard has been retained while adopting it in this Indian Standard.

CONTENTS

Page

SECTION 1: GENERAL

Clause

1 Scope and object ... 5
2 Definitions ... 7
3 Interface system overview ... 10

SECTION 2: FUNCTIONAL SPECIFICATIONS

4 Functional partition ... 14
5 Notation used to specify interface functions ... 19
6 Source handshake interface (SH) function ... 22
7 Acceptor handshake interface (AH) function ... 26
8 Talker interface (T) function (includes serial poll capabilities) ... 30
9 Listener interface (L) function ... 38
10 Service request interface (SR) function ... 43
11 Remote/local interface (RL) function ... 46
12 Parallel poll interface (PP) function ... 49
13 Device clear interface (DC) function ... 53
14 Device trigger interface (DT) function ... 55
15 Controller interface (C) function ... 57
16 Remote message coding and transfer ... 69

SECTION 3: ELECTRICAL SPECIFICATIONS

17 Application ... 74
18 Logical and electrical state relationships ... 74
19 Driver requirements ... 75
20 Receiver requirements ... 75
21 Composite device load requirements ... 76
22 Ground requirements ... 78
23 Cable characteristics ... 78
24 State transition time value ... 79

SECTION 4: MECHANICAL SPECIFICATIONS

25 Application ... 80
26 Connector type ... 81
27 Connector contact assignment ... 82
28 Device connector mounting ... 82
29 Cable assembly ... 83

SECTION 5: SYSTEM APPLICATIONS AND GUIDELINES FOR THE DESIGNER

30 System compatibility ... 85
31 Data rate consideration ... 86
32 Device capabilities ... 87
33 AND and OR logic operations ... 88
34 Address assignment ... 90
35 Typical combinations of interface functions ... 90
36 Unimplemented interface message handling ... 91

Clause Page

SECTION 6: SYSTEM REQUIREMENTS AND GUIDELINES FOR THE USER

37 System compatibility 92
38 System installation requirements 92
39 Address assignment 93
40 Cabling restrictions 94
41 Operational sequences 95

Tables

I Interface function repertoire 16
II SH mnemonics 23
III SH message output 24
IV Allowable subsets to the SH function 26
V AH mnemonics 27
VI AH message outputs 27
VII Allowable subsets to the AH function 29
VIII T mnemonics 31
IX T or TE message outputs 32
X TE mnemonics 33
XI Allowable subsets to the talker function 36
XII Allowable subsets to the extended talker function 37
XIII L mnemonics 38
XIV L or LE message output 38
XV LE mnemonics 39
XVI Allowable subsets for listener function 42
XVII Allowable subsets for extended listener function 43
XVIII SR mnemonics 45
XIX SR message output 45
XX Allowable subsets for Service Request interface function 45
XXI RL mnemonics 48
XXII RL message outputs 48
XXIII Allowable subsets for Remote/Local function 48
XXIV P mnemonics 51
XXV PP message outputs 51
XXVI Parallel poll message 52
XXVII Allowable subsets for Parallell poll interface function 53
XXVIII DC mnemonics 55
XXIX DC message output 55
XXX Allowable subsets for device clear interface function 55
XXXI DT mnemonics 56
XXXII DT message outputs 57
XXXIII Allowable subsets for device trigger interface function 57
XXXIV C mnemonics 59
XXXV C message output 60
XXXVI 61
XXXVII Controller function allowable subsets 68
XXXVIII Remote message coding 72
XXXIX Time values 80

Clause Page

Figures

1 Interface capacities and bus structure 12
2 Partition of functions within a device 15
3 Source handshake state diagram 23
4 Acceptor handshake state diagram 27
5 Talker state diagram 31
6 Extended talker state diagram 32
7 Listener state diagram 38
8 Extended listener state diagram 39
9 Service request state diagram 44
10 Renote local state diagram 47
11 Parallel poll state diagram 50
12 Device clear state diagram 54
13 Device trigger state diagram 56
14 Controller state diagram 58
15 DC load boundary specification 77
16 Typical signal line input/output circuits 77
17 Connector: mechanical requirements 81
18 Connector panel cut-out 82
19 Stud mount dimensions 83
20 Locking screw 84
21 Cable connector housing 85
22 Remote/local message paths 88
23 Bi-state signal line logic (open collector drivers) 90
24 Signal line logic and timing relationship 91

Annexes

A Typical instrument system 98
B Handshake process timing sequence 100
C Interface function allowable subset reference list 104
D Interface message reference list 110
E Multiline interface messages: ISO code representation 112
F Logic circuit implementation 113
G Parallel poll sequence 115
H Description of interface parameters on data sheets 116
J Recommended methods for improving the screening of cables that are specified in this part 122
K Information on the 25-pin connector 125
L Address switch labelling and interface status indicators 130

As in the Original Standard, this Page is Intentionally Left Blank

Indian Standard

PROGRAMMABLE MEASURING INSTRUMENTS — INTERFACE SYSTEM (BYTE SERIAL, BIT PARALLEL)

PART 1 FUNCTIONAL, ELECTRICAL AND MECHANICAL SPECIFICATIONS, SYSTEM APPLICATIONS AND REQUIREMENTS FOR THE DESIGNER AND USER

(First Revision)

SECTION 1: GENERAL

1 Scope and object

1.1 *Scope*

1.1.1 This part of IEC 625 is applicable to interface systems used to interconnect both programmable and non-programmable electronic measuring apparatus with other apparatus and accessories necessary to assemble instrumentation systems.

1.1.2 This part is applicable to the interface of instrumentation systems, or portions of them, in which:

- the data exchanged among the interconnected apparatus is digital (as distinct from analogue);
- the number of devices that may be interconnected by one contiguous bus does not exceed 15;
- the total transmission path length over the interconnecting cables does not exceed 20 m;
- the data rate across the interface on any signal line does not exceed 1 megabit per second.

The basic functional specifications of this part may be used in digital interface applications which require longer distances, more devices, increased noise immunity, or combinations of these. Different electrical and mechanical specifications may be required (e.g. symmetrical circuit configurations, high-threshold logic, special connectors or cable configurations) for these extended applications.

1.1.3 This part may also be applicable to other instrumentation system elements such as processors, stimulus, display, or storage devices, and terminal units found useful in instrumentation systems.

1.1.4 This part applies generally to laboratory and production test environments which are both electrically quiet and restricted as to physical dimensions (distances between the system components).

1.1.5 A primary focus of this part is to set forth an interface system to interconnect self-contained apparatus to other apparatus by external means. This standard may be also applied to interconnect the internal subsections within a self-contained equipment.

1.2 *Object*

This part is intended:

- to define a general-purpose system for use in limited distance applications,
- to specify the device-independent mechanical, electrical, and functional interface requirements which apparatus shall meet in order to be interconnected and communicate unambiguously via the system,
- to specify the terminology and definitions related to the system,
- to enable the interconnection of independently manufactured apparatus into a single functional system,
- to permit apparatus with a wide range of capability, from the simple to the complex, to be interconnected to the system simultaneously,
- to permit direct communication between apparatus without requiring all messages to be routed to a control or intermediate unit,
- to define a system with a minimum of restrictions on the performance characteristics of apparatus connected to the system,
- to define a system that permits asynchronous communication over a wide range of data rates,
- to define a system which, of itself, may be relatively low-cost and permits the interconnection of low-cost devices,
- to define a system that is easy to use.

1.2.1 This part deals only with the interface characteristics of instrumentation systems to the exclusion of design specifications, consideration of radio-interference regulations, performance requirements, and safety requirements of apparatus.

NOTE - For the latter two items, reference is made to IEC 348: *Safety requirements for electronic measuring apparatus*, and IEC 359: *Expression of the functional performance of electronic measuring equipment*.

1.3 *Normative references*

The following normative documents contain provisions which, through reference in this text, constitute provisions of this part of IEC 625. At the time of publication, the editions indicated were valid. All normative documents are subject to revision, and parties to agreements based on this part of IEC 625 are encouraged to investigate the possibility of applying the most recent editions of the normative documents indicated below. Members of IEC and ISO maintain registers of currently valid International Standards.

IEC 50(351): 1975, *International Electrotechnical Vocabulary – Chapter 351: Automatic control*

IEC 68-1: 1988, *Environmental testing – Part 1: General and guidance*

IEC 348: 1978, *Safety requirements for electronic measuring apparatus*

IEC 359: 1987, *Expression of the performance of electrical and electronic measuring equipment*

IEC 512-6: 1984, *Electromechanical components for electronic equipment; basic testing procedures and measuring methods – Part 6: Climatic tests and soldering tests*

IEC 603-2: 1988, *Connectors for frequencies below 3 MHz for use with printed boards – Part 2: Two-part connectors for printed boards, for basic grid of 2,54 mm (0,1 in) with common mounting features*

IEC 625-2: 1993, *Programmable measuring instruments – Interface system (byte serial, bit parallel) – Part 2: Codes, formats, protocols and common commands*

ISO/IEC 646: 1991, *Information technology – ISO 7-bit coded character set for information interchange*

IEC 807-2: 1992, *Rectangular connectors for frequencies below 3 MHz – Part 2: Detail specification for a range of connectors, with assessed quality, with trapezoidal shaped metal shells and round contacts – Fixed solder contact types*

2 Definitions

For the purposes of this part of IEC 625, the following definitions apply.

Throughout this part, and in so far as further distinction is not necessary:

- the term "system" denotes the byte-serial, bit-parallel interface system that, in general, includes all circuits, cables, connectors, message repertoire, and control protocol to effect unambiguous data transfer between devices;

- the term "device" or "apparatus" denotes any programmable measurement device or other product connected to the interface system that communicates information via and conforms to, the interface system definition.

This clause contains only general definitions. Detailed definitions are given under the headings of further clauses and subclauses, as appropriate.

2.1 *General system terms*

2.1.1 *System* (from International Electrotechnical Vocabulary (IEV 351-01-01)

A set of interconnected elements constituted to achieve a given objective by performing a specified function.

NOTE - The system is considered to be separated from the environment and other external systems by an imaginary surface which cuts the links between them and the considered system. Through these links, the system is affected by the environment, is acted upon by external systems, or acts itself on the environment or the external systems.

2.1.2 *Interface*

A common boundary between a considered system and another system, or between parts of a system, through which information is conveyed.

2.1.3 *Interface system*

The set of device-independent mechanical, electrical and functional elements of an interface necessary to effect communication among a set of devices. Cables, connectors, driver and receiver circuits, signal-line descriptions, timing and control conventions and functional logic circuits are typical system elements.

2.1.4 *Programmable*

That characteristic of a device that makes it capable of accepting data to alter the state of its internal circuits to perform a *specific* tasks(s).

2.1.5 *Remote control*

A method whereby a device is programmable via its electrical interface connection in order to enable the device to perform different tasks.

2.1.6 *Local control*

A method whereby a device is programmable by means of its local (front or rear panel) control in order to enable the device to perform various tasks. (Also referred to as *manual control.)*

2.1.7 *Compatibility*

The degree to which devices may be interconnected and used, without modification, when designed as defined throughout this part (e.g. mechanical, electrical, functional).

2.1.8 *Handshake cycle*

The process whereby digital signals effect the transfer of each data character across the interface by means of an *interlocked sequence* of status and control signals. *Interlocked* denotes a fixed sequence of events in which one event in the sequence must occur before the next event may occur.

2.1.9 *Programmable measuring apparatus*

A measuring apparatus that performs specified operations on command from the system and may transmit the results of the measurement(s) to the system.

2.1.10 *Terminal unit*

An apparatus that terminates the considered interface system and by means of which a connection (including code translation, if required) is made between the considered interface system and another external interface system.

2.2 *Signals and paths*

2.2.1 *Signal*

The physical representation of information.

NOTE - For the purpose of this part, this is a restricted definition of what is often called "signal" in the general sense, and hereinafter refers to digital electrical signals only.

2.2.2 *Signal parameter*

That parameter of an electrical quantity whose value or sequence of values conveys information.

2.2.3 *Signal level*

The magnitude of a signal compared to an arbitrary reference magnitude (voltage in the case of this part).

2.2.3.1 *High state*

The relatively more positive signal level used to assert a specific message content associated with one of two binary logic states.

2.2.3.2 *Low state*

This relatively less positive signal level used to assert a specific message content associated with one of two binary logic states.

2.2.4 *Signal line*

One of a set of signal conductors in an interface system used to transfer messages among interconnected devices.

2.2.5 *Byte*

A group of concurrent binary digits operated on as a unit and usually shorter than a computer word. (Frequently connotes a group of eight bits.)

2.2.6 *Bus*

A signal line or a set of signal lines used by an interface system to which a number of devices are connected and over which messages are carried.

2.2.6.1 *Unidirectional bus*

A bus used by any individual device for one-way transmission of messages only, i.e. either input only or output only.

2.2.6.2 *Bidirectional bus*

A bus used by any individual device for two-way transmission of messages, i.e. both input and output.

2.2.7 *Byte serial*

A sequence of bit-parallel data bytes used to carry information over a common bus.

2.2.8 *Bit parallel*

A set of concurrent data bits present on a like number of signal lines used to carry information. Bit-parallel data bits may be acted upon concurrently as a group (byte) or independently as individual data bits.

3 Interface system overview

3.1 *Interface system objective*

3.1.1 The overall purpose of an interface system is to provide an effective communication link over which *messages* are carried in an unambiguous way among a group of interconnected devices.

3.1.2 Messages (quantities of information) carried by an interface system belong to either of two broad categories:

- messages used to manage the interface system itself, hereinafter called *interface messages;*
- messages used by the devices interconnected via the interface system that are carried by, but not used or processed by, the interface system directly, hereinafter called *device-dependent messages.*

NOTE - The detailed specification on device-dependent messages is beyond the scope of this part.

3.2 *Fundamental communication capabilities*

3.2.1 An effective communication link requires three basic functional elements to organize and manage the flow of information to be interchanged among devices:

- a device acting as a *listener;*
- a device acting as a *talker;*
- a device acting as a *controller.*

NOTE - The terms *talker* and *listener* have specialized meanings in this part to better describe the role of devices. These terms are used in order to avoid confusion with other technical terms such as transmitter and receiver.

3.2.2 In the context of the interface system described by this part:

a) A device with the capability to listen can be addressed by an interface message to receive device-dependent messages from another device connected to the interface system.

b) A device with the capability to talk can be addressed by an interface message to send device-dependent messages to another device connected to the interface system.

c) A device with the capability to control can address other devices to listen or to talk. In addition, this device can send interface messages to command specified actions within other devices. A device with only this capability neither sends nor receives device-dependent messages.

NOTE - The use of the word controller throughout this part applies strictly to the management (control) of the interface system and does not imply the broad capabilities typically associated with the word in the data processing context. Further classification of the controller will be made in section 2 to distinguish between different types of controller capabilities used in the interface system.

3.2.3 Listener, talker, and controller capabilities occur individually or in any combination in devices interconnected via the interface system as shown in figure 1.

3.2.4 In addition to the basic listener, talker and controller functions, the system provides interface messages to accomplish the following operations:

a) A *serial poll* sequence may be initiated when a device (with *talker* function) requires some action by the controller, by transmitting the *service request* message. The controller will then obtain the status byte of all possible devices in sequence to ascertain which requires service.

b) The *parallel poll* function provides a device with the ability to transmit, on the controller's demand, one bit of status information (*request* service) simultaneously with several other devices. The assignment of a data line to a particular device for the response to a parallel poll may be accomplished through interface messages.

c) The *device-clear* and *device-trigger* functions provide a device with the ability to be initialized or triggered, on command from the controller. This may occur simultaneously with either selected devices or all devices in a system.

d) The *remote/local* function provides a device with the ability to accept program data from the bus, local data (for example, front panel controls), or both.

3.3 *Message paths and bus structure*

3.3.1 The interface system contains a set of 16 signal lines used to carry all information, interface messages and device-dependent messages, among interconnected devices.

3.3.2 Messages may be coded on one signal line or on a set of signal lines as determined by the particular message content and its relationship to the interface system.

3.3.3 The bus structure is organized into three sets of signal lines:

- data bus, eight signal lines;
- data byte transfer control bus, three signal lines;
- interface management bus, five signal lines.

Figure 1 illustrates the basic communication paths.

3.3.3.1 A set of eight interface signal lines carries all 7-bit interface messages and the device-dependent messages:

a) DATA INPUT OUTPUT 1 (DIO1);

•   •
•   •
•   •

h) DATA INPUT OUTPUT 8 (DIO8).

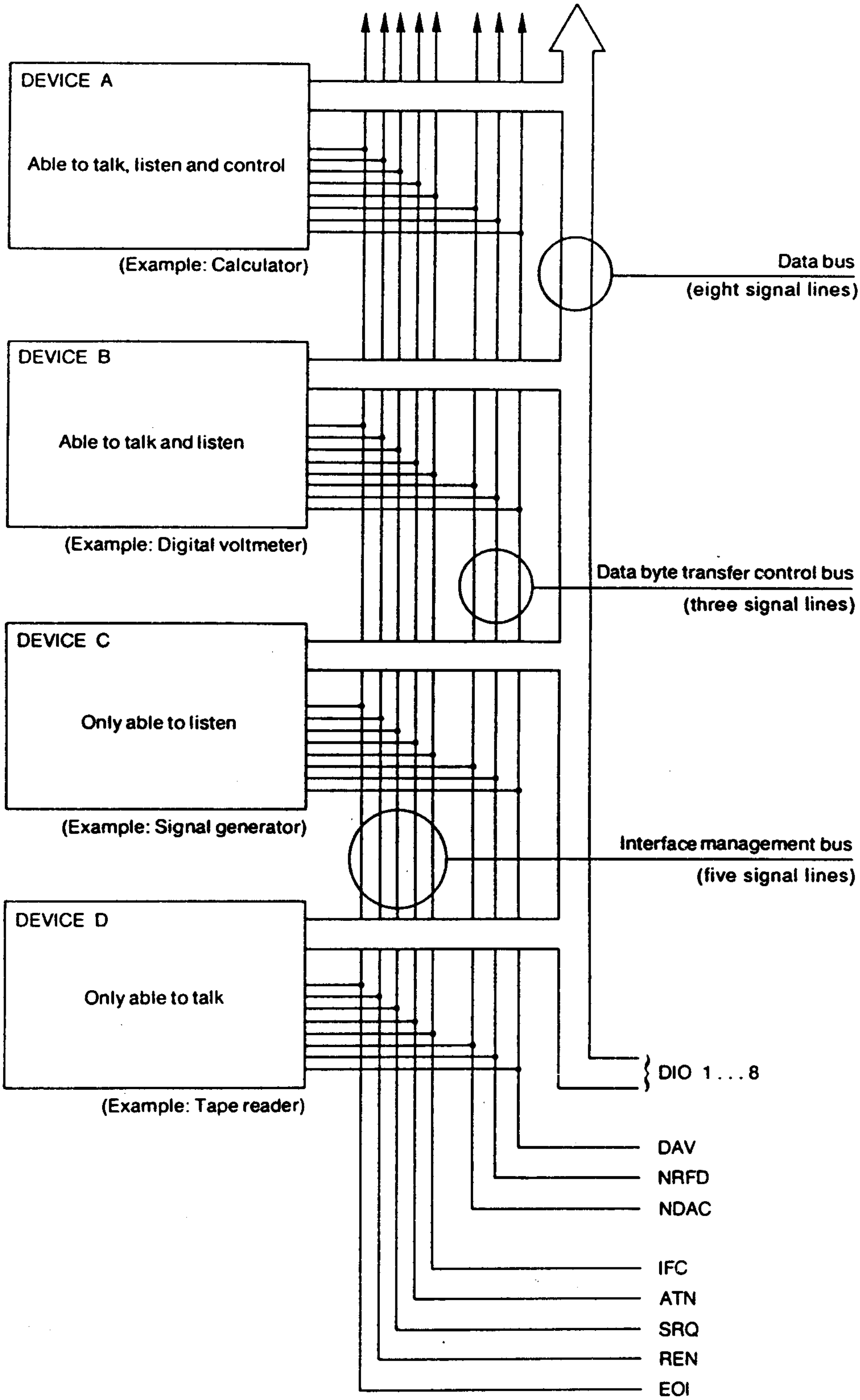


Figure 1 – Interface capabilities and bus structure

Message bytes are carried on the DIO signal lines:

- in a bit-parallel byte-serial form;
- asynchronously;
- bidirectionally.

NOTE - A message may be carried on an individual DIO signal line when required.

3.3.3.2 A set of three signal lines is used to effect the transfer of each byte of data on the DIO signal lines from a talker or controller to one or more listeners:

a) DATA VALID (DAV) is used to indicate the condition (availability and validity) of information on the DIO signal lines.

b) NOT READY FOR DATA (NRFD) is used to indicate the condition of readiness of device(s) to accept data.

c) NOT DATA ACCEPTED (NDAC) is used to indicate the condition of acceptance of data by device(s).

The DAV, NRFD, and NDAC signal lines operate in what is called a three-wire (interlocked) handshake process to transfer each data byte across the interface.

3.3.3.3 Five interface signal lines are used to manage an orderly flow of information across the interface:

a) ATTENTION (ATN) is used (by a controller) to specify how data on the DIO signal lines are to be interpreted and which devices must respond to the data.

b) INTERFACE CLEAR (IFC) is used (by a controller) to place the interface system, portions of which are contained in all interconnected devices, in a known quiescent state.

c) SERVICE REQUEST (SRQ) is used (by a device) to indicate the need for attention and to request an interruption of the current sequence of events.

d) REMOTE ENABLE (REN) is used (by a controller) in conjunction with other messages, to enable or disable one or more local controls that have corresponding remote controls.

e) END OR IDENTIFY (EOI) is used (by a talker) to indicate the end of a multiple byte transfer sequence or in conjunction with ATN (by a controller), to execute a parallel poll.

3.4 *Interface system elements*

The primary elements of this interface system are:

- functional elements;
- electrical elements;
- mechanical elements.

Each is described in a following section.

SECTION 2: FUNCTIONAL SPECIFICATIONS

4 Functional partition

A device is a physical entity designed for a particular application. It may be partitioned conceptually into three major functional areas each containing unique capabilities:

- device functions (definition is application-dependent);
- interface functions (definition is application-independent);
- message coding logic.

All communication to or from interface functions is defined in terms of messages and state linkages (see 4.3).

All messages carried on the signal lines are coded according to the coding logic defined in clause 16.

4.1 *Device functions*

The scope, purpose, size, content, and organization of the device function area (e.g. analogue signal measurement capability, range, modes of operation, etc.) are beyond the scope of this part. Figure 2 illustrates the device function area (B) for which the designer has complete freedom to define device-related capability and the interface function area (A) for which the designer has no freedom to define new capability beyond that specified in this part.

4.2 *Interface function concepts*

4.2.1 *Interface function*

An interface function is the system element that provides the basic operational facility through which a device can receive, process, and send messages. A number of interface functions, each of which acts in accordance with specific protocol, are defined throughout this section of the part. Each specific interface function may only send or receive a limited set of messages within each particular class of messages.

4.2.2 *Interface function state*

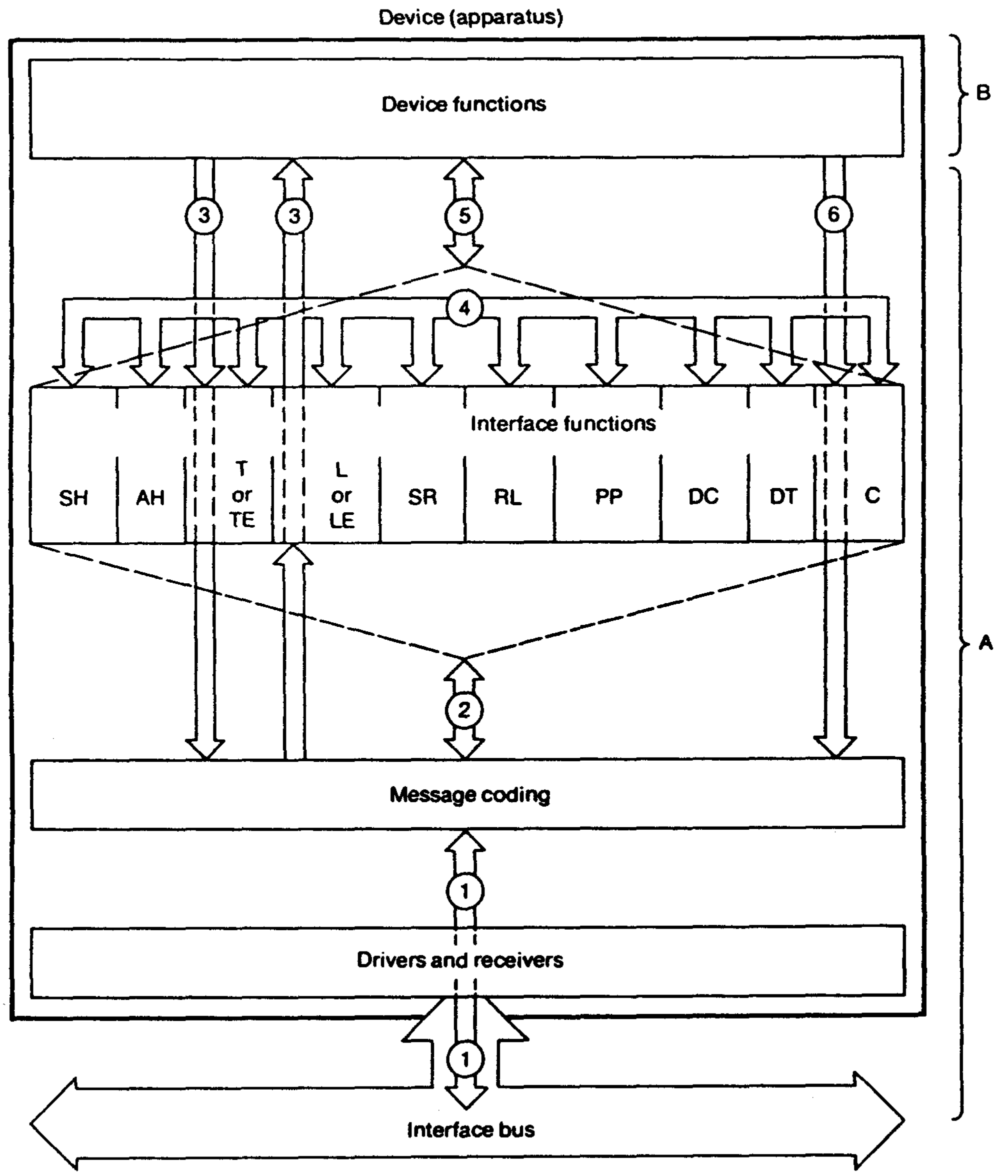
4.2.2.1 Each of the interface functions is defined in terms of one or more groups of interconnected, mutually exclusive *states*.

4.2.2.2 One and only one state shall be active at any one time within a single group of interconnected, mutually exclusive states.

4.2.2.3 For each state of an interface function definitions are given for:

- messages which may or must be sent over the interface while that state is active, and
- conditions under which the function must leave that state and enter one of the other states in its group.

These messages and conditions define the processing capability of the state.



A is the capability defined by this part

B is the capability defined by the designer

1 are the interface bus signal lines

2 are the remote interface messages to and from interface functions

3 are the device-dependent messages to and from device functions

4 are the state linkages between interface functions

5 are the local messages between device functions and interface functions (messages to interface functions are defined in this part; messages from interface functions are specified by the designer)

6 are the remote interface messages sent by device functions within a controller

Figure 2 – Partition of functions within a device

4.2.3 *Interface function repertoire*

4.2.3.1 The designer is given the choice to select the particular set of interface functions necessary to fit the particular device application area. Figure 2 and table I identify the available interface functions.

TABLE I

Interface function repertoire		
Interface function	Symbol	Relevant message path
Source handshake	SH	1, 2, 4, 5
Acceptor handshake	AH	1, 2, 4, 5
Talker or extended talker	T or TE	1, 2, 3, 4, 5
Listener or extended listener	L or LE	1, 2, 3, 4, 5
Service request	SR	1, 2, 4, 5
Remote local	RL	1, 2, 4, 5
Parallel poll	PP	1, 2, 4, 5
Device clear	DC	1, 2, 4, 5
Device trigger	DT	1, 2, 4, 5
Controller	C	1, 2, 4, 5, 6

4.2.3.2 The total processing capability of a set of interface functions (designer-selected set included in a specific device) at any moment is the logic conjunction of the processing capabilities of all those states (within each individual interface function) that are active at that moment.

4.2.4 *Interface function assumptions and perspective*

4.2.4.1 The state diagrams used to define the interface functions do not indicate, either explicity or implicity, the intended existence of specific circuit elements to achieve the logical and physical implementation of a function; for example, not all states necessarily imply the existence of a latched flip-flop or other memory element.

4.2.4.2 The state diagrams used to define the interface functions are intended to permit the use of a wide variety of logic circuit implementation (e.g. random logic, sequential logic, etc.).

4.2.4.3 The designer is free to combine and implement two or more interface functions with one logic design, provided all the conditions for each state of each interface function as defined in this section are met.

4.2.4.4 Throughout this section of the part, the state diagrams, written descriptions, requirements, and guidelines are written for and should be interpreted from the *device perspective.* Sections 5 and 6 will describe the interaction among devices from the *system perspective.*

4.2.4.5 An interface function shall ignore (not respond to) any message coding not specifically defined.

4.2.4.6 A function may stay in any state for any amount of time (including zero) after exit conditions are met if this is not in conflict with specified constraints.

4.3 *Message concepts*

4.3.1 *Message*

Each message represents a quantity of information and will be received either true or false at any specific time.

All communication between an interface function and its environment is accomplished through messages sent or received.

4.3.2 *Local message route and content*

4.3.2.1 Messages sent between a device function and an interface function are called *local messages.*

4.3.2.2 Local messages flow between device functions and interface functions (see figure 2, message route 5).

NOTE - Certain local messages are conveyed as remote messages and vice versa.

4.3.2.3 The designer is not allowed to introduce new local messages to interface functions.

4.3.2.4 The designer is allowed to introduce a local message derived from any state of any interface function to device function(s).

4.3.2.5 Local messages sent by device functions shall exist for enough time to cause the required state transitions.

4.3.3 *Remote message route and content*

4.3.3.1 Messages sent via the interface between interface functions of different devices are called *remote messages.*

4.3.3.2 Each remote message is either an *interface message* or a *device-dependent message.*

4.3.3.3 Each interface message is sent to cause a state transition within another interface function. An interface message will not be passed along to the device when received by an interface function (as shown in figure 2, message route 2).

4.3.3.4 Device-dependent messages are passed between the device functions and the message-coding logic via specified interface functions. These will cause no state transitions within the interface functions. Examples of device-dependent messages include device programming data, device measurement data, and device status data (as shown in figure 2, message route 3).

4.3.4 *State linkage route and content*

4.3.4.1 A state linkage is the logical interconnection of two interface functions where the transition to an active state of one interface function is dependent on the existence of a specified active state of another interface function as indicated in figure 2, message route 4.

4.3.5 *Message coding*

4.3.5.1 Message coding is the act of translating remote messages to or from interface signal-line values.

4.3.5.2 A message sent over a single signal line is called a *uniline message.* Two or more of these messages can be sent concurrently.

4.3.5.3 A message which shares a group of signal lines with other messages, in some mutually exclusive set, is called a *multiline message.* Only one multiline message (message byte) can be sent at one time.

4.3.6 *Classification of multiline messages*

4.3.6.1 Multiline messages are interpreted as interface messages when the ATN message is true.

4.3.6.2 Multiline messages are interpreted as device-dependent messages when the ATN message is false.

4.3.6.3 The ATN message, when true, enables the accepting and processing of these specific classes of multiline messages:

- universal commands (all devices independent of interface function states);
- addressed commands (all devices addressed to listen);
- addresses (all devices);
- secondary addresses or commands (all devices enabled by a primary address or command).

For a list of specific commands, see table XXXVI.

4.3.6.4 Each device, when the ATN message is false, may accept and process its own device-dependent message when addressed to listen.

4.3.7 *Message transfer conventions*

4.3.7.1 *Remote message transfer conventions*

a) The value (true or false) of all remote messages capable of being sent by a device shall at all times be as dictated by active states of its interface functions.

b) The interface signal line(s) used to send a message value shall be set to the levels specified by the remote message coding table XXXVIII.

c) Since normal interface operation allows two or more devices to simultaneously send opposite values of the remote messages, a technique must be provided for resolving these conflicts. This is accomplished by implementing two types of message transfer over the interface, *active* transfer and *passive* transfer. The interface is structured so that in all conflicts between two message values, one of them will be active and the other passive. Messages must be transferred in such a way that the active value *overrides* the passive value in every conflict that arises.

d) A remote message can be transferred in one of four ways:

- An *active true* value being sent is guaranteed to be the value received (the device need not allow it to be overridden).
- A *passive true* value being sent is not guaranteed to be the value received (the device must allow it to be overridden).
- An *active false* value being sent is guaranteed to be the value received (the device need not allow it to be overridden).
- A *passive false* value being sent is not guaranteed to be the value received (the device must allow it to be overridden).

e) Throughout the text, the terms "true" and "false" if not qualified are assumed to mean "active true" and "active false" during all discussions of remote message values sent by an interface function.

f) For two specific remote messages, DAC and RFD, only false values are defined to be sent actively. Thus, an AND operation can be considered to be performed on the interface signal lines (see clause 33).

g) For one remote message, SRQ, only true values are defined to be sent actively. Thus, an OR operation can be considered to be performed on the interface signal line (see clause 33).

h) Only the multline message(s) to be sent true will be specified for an interface function state since multiline messages (sent via the DIO lines) are by their nature mutually exclusive. It should be understood that all unspecified multiline messages are sent passive false while that state is active.

4.3.7.2 *Local message transfer conventions*

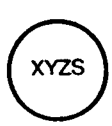
a) The coding of local messages is beyond the scope of this part and is left to the discretion of the device designer.

b) It is recommended that local messages qualifying transitions within any group of mutually exclusive states of an interface function be themselves mutually exclusive.

5 Notation used to specify interface functions

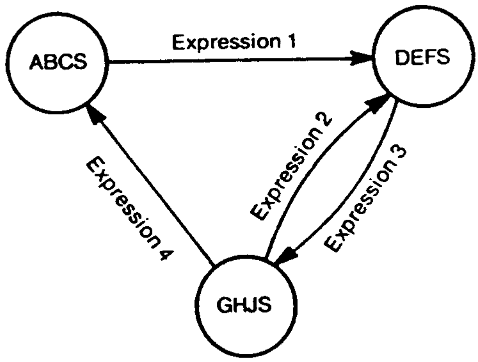
5.1 *State diagram notation*

5.1.1 Each state that an interface function can assume is represented graphically as a circle. A four-letter upper-case mnemonic, always ending in an S, is used within the circle to identify the state.



5.1.2 All permissible *transitions* between states of an interface function are represented graphically by arrows between them.

5.1.3 Each transition is qualified by an *expression* whose value is either true or false. The interface function must remain in its current state if all expressions which qualify transitions leading to other states are false. The interface function must enter the state pointed to if, and only if, one of these expressions becomes true. The new state may be entered at any time after the expression(s) become(s) true, unless a time value is specified.



5.1.3.1 An expression consists of one or more local messages, remote messages, state linkages, or minimum time limits used in conjunction with the operators AND, OR, or NOT.

5.1.3.2 A local message to an interface function is represented by a three-letter mnemonic written in lower case; for exemple, rdy. (Within the text, local message mnemonics are written in italics for clarity: e.g. *rdy*.)

5.1.3.3 A remote message (received via the interface) is represented by a three-letter mnemonic written in upper case: for example ATN.

5.1.3.4 A linkage from another state diagram is represented by a four-letter mnemonic enclosed in an oval: for example (LACS). A state linkage is true if the enclosed state is currently active; otherwise, it is false.

5.1.3.5 A minimum time-limit is represented by the symbol "*Tn*". This symbol achieves a true value only after the interface has been in the state originating the corresponding transition for the time value specified. It will remain true until the state is exited. The values for these time-limits are contained in table XXXIX.

5.1.3.6 The "AND" operator is represented by the symbol "$\wedge$".

5.1.3.7 The "OR" operator is represented by the symbol "$\vee$".

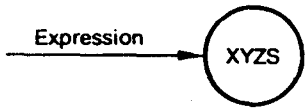
5.1.3.8 The "AND" operator takes precedence over the "OR" operator within an expression unless otherwise specified by parentheses.

5.1.3.9 The "NOT" operator is represented by a horizontal bar placed over the portion of the expression to be negated. The resulting negated expression has a true value if, and only if, the value of the expression under the bar is false.

5.1.4 If a transition is further qualified by a *maximum time-limitf* "(within *tn*)", then the state pointed to must be entered within the specified amount of time after the expression becomes true. The value for these time-limits are contained in table XXXIX.

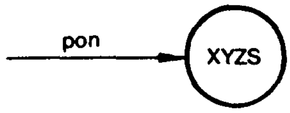
5.1.5 If a portion of an expression is optional in that its true value is not required for the complete expression to be true (at the designer's choice), then it is enclosed within square brackets "[...]".

5.1.6 If a specific expression causes a transition to a state from all other states of the diagram, a shorthand notation is used instead of all the individual transitions being drawn. An arrow without a state at its origin is used to represent this condition, and is assumed to originate in all states (i.e. IFC or $\overline{\text{REN}}$). It is further assumed that these same expressions are false (i.e. $\overline{\text{IFC}}$ or REN) to permit all other transitions in the diagram, and therefore they are omitted from the diagrams.

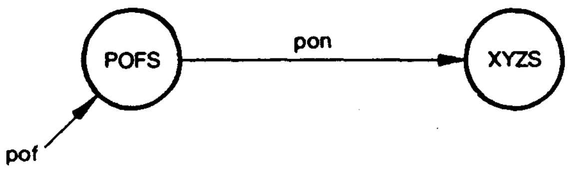


5.1.7 Although "power-off" (POFS) is a valid state of most interface functions and should normally be shown on all diagrams with a transition leading to the state to be entered at "power-on" time, a shorthand form is used showing the "pon" pseudo-message originating a transition to the first state to be entered when power is turned on.

– abbreviated notation used on state diagram:



– complete representation implied by above symbol:



5.2 *Message output notation*

5.2.1 The message output table included with each interface function state diagram summarizes only the remote messages allowed to be sent during each of the states of the function.

5.2.2 Rows of the table are used to indicate states of the interface function.

5.2.3 Columns of the table are used to indicate remote messages allowed to be sent during at least one state of the interface function.

5.2.4 Each table entry indicates the value of a message that shall be sent while a specified state is active:

- "T" indicates active true.
- "F" indicates active false.
- "(T)" indicates passive true.
- "(F)" indicates passive false.

5.2.5 One column in each table is allocated, if required, to the group of multiline remote messages allowed to be sent. The multiline message to be sent true during each state is placed in its corresponding table entry. False values are not shown since multiline messages are mutually exclusive. Parentheses around a multiline message name specify that it must be sent passive rather than active true.

5.2.6 A separate column for device function interaction summarizes the corresponding types of messages (or resultant action) device functions are allowed to send or receive. Local messages beyond the scope of this part, from the interface function to the device functions may be used to coordinate the appropriate action at the choice of the designer.

6 Source handshake interface (SH) function

6.1 *General description*

The SH function provides a device with the capability to guarantee the proper transfer of multiline messages. An interlocked handshake sequence between the SH function and one or more Acceptor Handshake Interface Functions (each contained within separate devices) guarantees asynchronous transfer of each multiline message. The SH function controls the initiation of, and termination of, the transfer of a multiline message byte. This function utilizes the DAV, RFD, and DAC messages to effect each message byte transfer.

6.2 *Source handshake function state diagram*

The SH function shall be implemented so as to perform according to the state diagram given in figure 3 and the state descriptions given throughout clause 6. Table II specifies the set of messages and states required to effect transition from one active state to another. Table III specifies the messages that shall be sent and the device function interaction required while each state is active.

6.3 *Source handshake function state descriptions*

6.3.1 *SOURCE IDLE STATE (SIDS)*

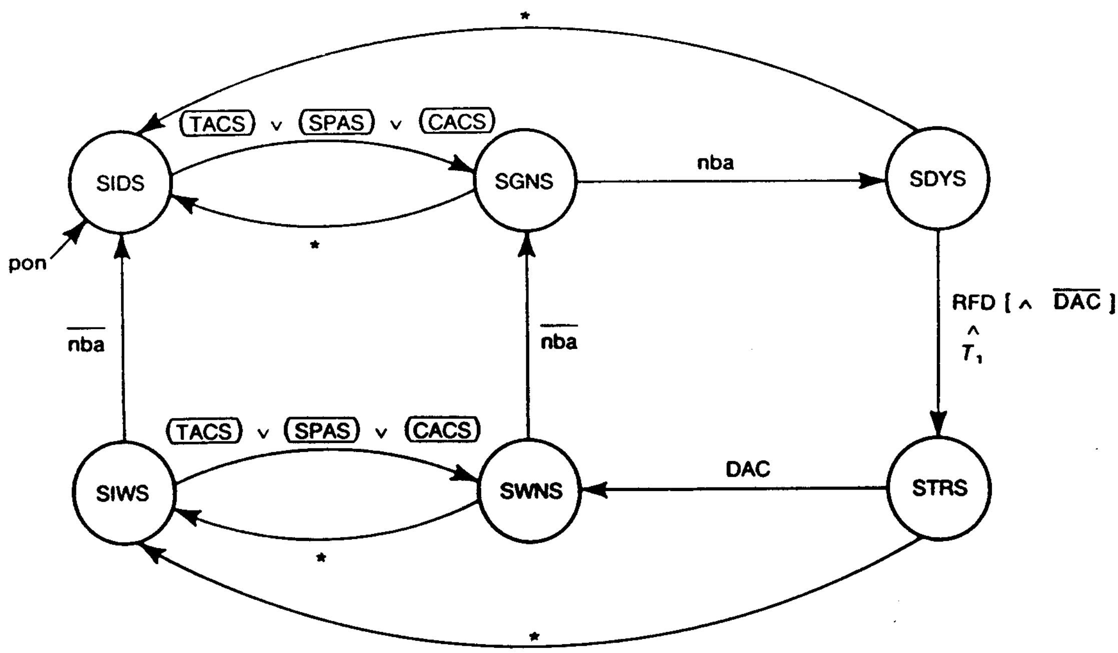
6.3.1.1 In the SIDS state, the SH function is not engaged in the handshake cycle and does not have a new message byte available.

The SH function powers on in the SIDS state.

6.3.1.2 In the SIDS state the DAV message shall be sent passive false.

6.3.1.3 The SH function shall exit the SIDS state and enter the SOURCE GENERATE STATE (SGNS) if:

- the TALKER ACTIVE STATE (TACS) is active;
- or the SERIAL POLL ACTIVE STATE (SPAS) is active;
- or the CONTROLLER ACTIVE STATE (CACS) is active.



$* \; (ATN \wedge \overline{CACS \vee CTRS}) \vee (\overline{ATN} \wedge \overline{TACS \vee SPAS})$

(within t_2)

Figure 3 – Source handshake state diagram

TABLE II – SH mnemonics

Messages:	**Interface states:**
pon – power on	SDYS – SOURCE DELAY STATE
nba – new byte available	STRS – SOURCE TRANSFER STATE
ATN – ATTENTION	SWNS – SOURCE WAIT FOR NEW CYCLE STATE
RFD – READY FOR DATA	SIWS – SOURCE IDLE WAIT STATE
DAC – DATA ACCEPTED	TACS – TALKER ACTIVE STATE
	SPAS – SERIAL POLL ACTIVE STATE (T FONCTION)
Interface states:	CACS – CONTROLLER ACTIVE STATE (C FUNCTION)
SIDS – SOURCE IDLE STATE	CTRS – CONTROLLER TRANSFER STATE (C FONCTION)
SGNS – SOURCE GENERATE STATE	

TABLE III – SH message outputs

SH state	Remote message sent DAV	Device function interaction (DF)
SIDS	(F)	Device function can change remote multiline messages
SGNS	F	Device function can change remote multiline messages
SDYS	F	DAB, EOS multiline and END messages shall not change
STRS	T	DAB, EOS multiline and END message shall not change
SWNS	T or F	Device function requested to change multiline messages
SIWS	(F)	Device function requested to change multiline messages

6.3.2 *SOURCE GENERATE STATE (SGNS)*

6.3.2.1 In the SGNS state the device is generating a new message byte and the function is waiting for the new byte to become available.

6.3.2.2 In the SGNS state the SH function shall send the DATA VALID (DAV) message false. In this state the device may change the multiline message being sent via the Talker or Controller Interface Function while in the TACS or CACS states.

6.3.2.3 The SH function shall exit the SGNS state and enter:

a) the SOURCE DELAY STATE (SDYS) if the *new byte available (nba)* message is true;

b) the SIDS state within t_2 if either:

– the ATTENTION (ATN) message is true and neither the CACS state nor the CTRS state is active;

– or the ATN message is false and neither the TACS state nor the SPAS state is active.

6.3.3 *SOURCE DELAY STATE (SDYS)*

6.3.3.1 In the SDYS state the SH function is waiting for a message byte to settle on the interface signal lines after the change during the SGNS state and for all the Acceptor Functions to indicate their readiness to accept the message byte.

6.3.3.2 In the SDYS state the SH function shall send the DAV message false. In this state the device shall not change the multiline message being sent.

6.3.3.3 The SH function shall exit the SDYS state and enter:

a) the SOURCE TRANSFER STATE (STRS) only after T_1 if the READY FOR DATA (RFD) message is true and the DAC message is false;

b) the SIDS state within t_2 if either:

– the ATTENTION (ATN) message is true and neither the CACS state nor the CTRS state is active;

– or the ATN message is false and neither the TACS state nor the SPAS state is active.

6.3.4 *SOURCE TRANSFER STATE (STRS)*

6.3.4.1 In the STRS state the SH function indicates to the acceptor handshake function that it is continuously sending a valid message byte.

6.3.4.2 In the STRS state the SH function shall send the DAV message true. In this state the device shall not change either the multiline message or the END message (if used) being sent.

6.3.4.3 The SH function shall exit the STRS state and enter:

a) the SOURCE IDLE WAIT STATE (SIWS) within t_2 if either:

   – the ATTENTION (ATN) message is true and neither the CACS state nor the CTRS state is active;

   – or the ATN message is false and neither the TACS state nor the SPAS state is active:

NOTE - This implies an asynchronous interrupt (see 15.5).

b) the SOURCE WAIT FOR NEW CYCLE STATE (SWNS) if the DATA ACCEPTED (DAC) message is true.

6.3.5 *SOURCE WAIT FOR NEW CYCLE STATE (SWNS)*

6.3.5.1 In the SWNS state the SH function is waiting for the device to start a new message generation cycle.

6.3.5.2 In the SWNS state the SH function may send the DAV message true or false. In this state the device may change the multiline message being sent.

6.3.5.3 The SH function shall exit the SWNS state and enter:

a) the SGNS state if the *nba* message is false;

b) the SIWS state within t_2 if either:

   – the ATTENTION (ATN) message is true and neither the CACS state nor the CTRS state is active;

   – or the ATN message is false and neither the TACS state nor the SPAS state is active.

6.3.6 *SOURCE IDLE WAIT STATE (SIWS)*

6.3.6.1 In the SIWS state the SH function is not active in the external message byte transfer process but is active in the internal process of waiting for the device to start a new message generation cycle. This SIWS state allows a sequence of message byte transfers to be interrupted without loss of data over the interface while at the same time the device may continue to prepare for the new (next) message byte generation cycle.

6.3.6.2 In the SIWS state the DAV message shall be sent passive false.

6.3.6.3 The SH function shall exit the SIWS state and enter:

a) the SIDS state if the *nba* message is false;

b) the SWNS state if either:
   - the TACS state is active;
   - or the SPAS state is active;
   - or the CACS state is active.

6.4 *Source handshake function allowable subsets*

The only allowable subsets to the SH function shall be those listed in table IV.

TABLE IV

Identification	Description	States omitted	Other requirements	Other function subsets required
SHØ	No capability	All	None	None
SH1	Complete capability	None	None	T1-T8 or TE1-TE8 or C5-C28

6.5 *Additional SH function requirements and guidelines*

6.5.1 *Interpretation of the local* nba *messages*

6.5.1.1 The *nba* true message indicates the device has generated a (new) message bye and made it available on the interface signal lines.

6.5.1.2 The *nba* message shall become true only in the SIDS or SGNS states. The *nba* message may become false in any other SH states.

6.5.2 *Alternate transition to idle states*

The expression $(\mathrm{ATN} \wedge \overline{\mathrm{CACS} \vee \mathrm{CTRS}}) \vee (\overline{\mathrm{ATN}} \wedge \overline{\mathrm{TACS} \vee \mathrm{SPAS}})$ for interruption may be substituted by $\overline{\mathrm{TACS}} \wedge \overline{\mathrm{SPAS}} \wedge \overline{\mathrm{CACS}} \wedge \overline{\mathrm{CTRS}}$ if the transition of the latter expression can be effected within t_2 after the change of ATN.

7 Acceptor handshake interface (AH) function

7.1 *General description*

The AH function provides a device with the capability to guarantee proper reception of remote multiline messages. An interlocked handshake sequence between an SH function and one or more AH functions (each contained within separate devices) guarantees asynchronous transfer of each message byte. An AH function may delay either the initiation of, or termination of, a multiline message transfer until prepared to continue with the transfer process. The AH function utilizes the DAV, RFD, and DAC messages to effect each message byte transfer.

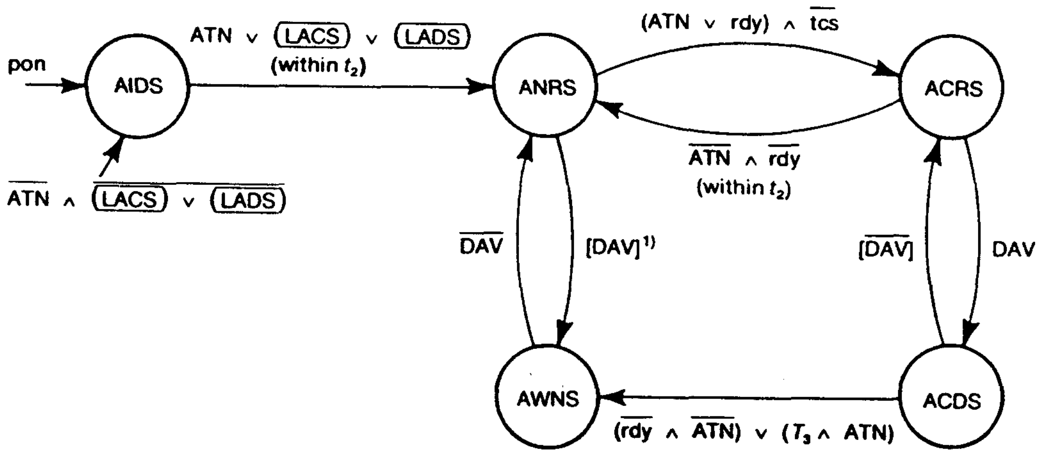
7.2 *Acceptor handshake function state diagram*

The AH function shall be implemented so as to perform according to the state diagram given in figure 4 and the state descriptions given throughout clause 7. Table V specifies the set of messages and states required to effect transition from one active state to another. Table VI specifies the messages that shall be sent and the device function interaction required while each state is active.

7.3 *Acceptor handshake function state descriptions*

7.3.1 *ACCEPTOR IDLE STATE (AIDS)*

7.3.1.1 In the AIDS state the AH function is inactive and not engaged in the handshake cycle. The AH function powers on in the AIDS state.



[1)] This transition will never occur under normal interface operation; however, it may be implemented to simplify the interface function design.

Figure 4 – Acceptor handshake state diagram

TABLE V – AH mnemonics

Messages:	Interface states:
pon – power on rdy – ready for next message tcs – take control synchronously[1)] ATN – ATTENTION DAV – DATA VALID	AIDS – ACCEPTOR IDLE STATE ANRS – ACCEPTOR NOT READY STATE ACRS – ACCEPTOR READY STATE ACDS – ACCEPT DATA STATE AWNS – ACCEPTOR WAIT FOR NEW CYCLE STATE (LADS) – LISTENER ADDRESSED STATE (L FUNCTION) (LACS) – LISTENER ACTIVE STATE (L FUNCTION)
[1)] See 15.3.7.1.	

TABLE VI – AH message outputs

AH state	Remote message sent		Device function interaction (DF)
	RFD	DAC	
AIDS	(T)	(T)	Device function can receive neither remote multiline nor END messages
ANRS	F	F	Device function can receive neither remote multiline nor END messages
ACRS	(T)	F	Device function can receive neither remote multiline nor END messages
AWNS	F	(T)	Device function can receive neither remote multiline nor END messages
ACDS	F	F	Device function can receive remote multiline messages if LACS is active

7.3.1.2 In the AIDS state the RFD and DAC messages shall be sent passive true.

7.3.1.3 The AH function shall exit the AIDS state and enter the ACCEPTOR NOT READY STATE (ANRS) within t_2 if either:

- the ATN message is true;
- or the LACS state is active;
- or the LADS state is active.

7.3.2 *ACCEPTOR NOT READY STATE (ANRS)*

7.3.2.1 In the ANRS state the AH function indicates to the interface it has not yet prepared internally to continue with the handshake cycle.

7.3.2.2 In the ANRS state the RFD and DAC messages shall be sent false.

7.3.2.3 The AH function shall exit the ANRS state and enter:

a) the ACRS state if the *take control synchronously (tcs)* message is false (see 15.3.7.1) and either:
   - the ATN message is true;
   - or the *rdy* message is true;

b) the AIDS state if the ATN message is false and neither:
   - the LADS state is active;
   - nor the LACS state is active;

c) the AWNS state if the DAV message is true (optional, note that this transition will never occur under normal interface operation).

7.3.3 *ACCEPTOR READY STATE (ACRS)*

7.3.3.1 In the ACRS state the AH function indicates to the interface that it is prepared to receive multiline messages.

7.3.3.2 In the ACRS state the DAC message shall be sent false and the RFD message shall be sent passive true.

7.3.3.3 The AH function shall exit the ACRS state and enter:

a) the ACCEPT DATA STATE (ACDS) if the DAV message is true;

b) the AIDS state if the ATN message is false and neither:
   - the LADS state is active;
   - nor the LACS state is active;

c) the ANRS state within t_2 if both the ATN and the *ready for next message (rdy)* messages are false.

7.3.4 *ACCEPT DATA STATE (ACDS)*

7.3.4.1 In the ACDS state the AH function indicates to the SH function that it shall maintain a valid message byte. This is the only state in which multiline messages on the DIO signal lines are valid. The ACDS state indicates to the interface functions that an interface message is present and valid if the ATN message is true. The ACDS state indicates to the device functions that a device-dependent message is present and valid if the LACS state is active.

7.3.4.2 In the ACDS state the DAC and RFD messages shall be sent false.

7.3.4.3 The AH function shall exit the ACDS state and enter:

a) the ACCEPTOR WAIT FOR NEW CYCLE STATE (AWNS) if either:

- the ATN message is true and a period of T_3 has elapsed;
- or the ATN and *rdy* messages are both false;

b) the AIDS state if the ATN message is false and neither:

- the LADS state is active;
- nor the LACS state is active;

c) the ACRS state if the DAV message is false (optionally, this transition can occur only when the controller takes control asynchronously).

7.3.5 *ACCEPTOR WAIT FOR NEW CYCLE STATE (AWNS)*

7.3.5.1 In the AWNS state the AH function indicates that it has received a multiline message byte.

7.3.5.2 In the AWNS state the RFD message shall be sent false and the DAC message shall be sent passive true.

7.3.5.3 The AH function shall exit the AWNS state and enter:

a) the ACCEPTOR NOT READY STATE (ANRS) if DAV is false;

b) the AIDS state if the ATN message is false and neither:

- the LADS state is active;
- nor the LACS state is active.

7.4 *Acceptor function allowable subsets*

The only allowable subsets to the AH function shall be those listed in table VII.

TABLE VII

Identification	Description	States omitted	Other requirements	Other function subsets required
AHØ	No capability	All	None	None
AH1	Complete capability	None	None	None

7.5 *Additional AH function requirements and guidelines*

7.5.1 The local message *rdy* shall not become false during the ACRS state. The transition from ACRS to ANRS shall occur only at the time ATN becomes false.

7.5.2 The RFD message received by an SH function is the logical AND of all the RFD messages sent by all the active AH functions. Similarly, the DAC message received by an SH function is the logical AND of all the DAC messages sent by all the AH functions. The way in which the composite effects of multiple AH functions interact with an SH function to perform the logical AND function via the use of the NRFD and NDAC signal lines is explained further in clause 33.

7.5.3 Since interface functions need be designed only "so as to perform according to" the state diagrams specified, it is not required that exactly the states specified are the ones which exist in an implementation. One consequence of this statement is that interface function state transitions which are qualified by interface messages can occur *after* the message has been received as long as the RFD message is held false until they occur. The resulting performance cannot be distinguished from the performance of the specified diagrams in which the transitions must occur *while* the interface message is being received. If this type of implementation is chosen, then the AH function should remain in the ANRS state even though the exit condition is true in order to hold the RFD message false (this is allowed by 4.2.4.6).

8 Talker interface (T) function (includes serial poll capabilities)

8.1 *General description*

8.1.1 The T function provides a device with the capability to send device-dependent data (including status data during a serial poll sequence) over the interface to other devices. The capability exists only when the T function is addressed to talk.

8.1.2 There are two alternative versions of the functions: one with and one without address extension. The normal T function uses a 1-byte address, the primary talk address. The T interface function with address extension (hereinafter called the extended talker (TE) function) uses a 2-byte address, the primary and secondary talk addresses. In all other respects, the capabilities of both versions are tne same.

8.1.3 Only one of the two alternative talker functions need be implemented in a specific device.

NOTE - Both the T function and TE function are described concurrently throughout clause 8 due to the extensive similarity between these two functions.

8.2 *Talker function state diagrams*

8.2.1 The T function shall be implemented so as to perform according to the state diagrams given in figure 5 and the state descriptions given throughout clause 8. Table VIII specifies the set of messages and states required to effect transition from one active state to another. Table IX specifies the messages that shall be sent and the device function interaction required while each state is active.

8.2.2 The TE function shall be implemented so as to perform according to the state diagrams given in figure 6 and the state descriptions given throughout clause 8. Table X specifies the set of messages and states required to effect transition from one active state to another. Table IX specifies the messages that shall be sent and the device function interaction required while each state is active.

8.3 *Talker function state descriptions*

8.3.1 *TALKER IDLE STATE (TIDS)*

8.3.1.1 In the TIDS state either the T function or the TE function is not engaged in sending data or status bytes. The T function or the TE function powers on in the TIDS state.

8.3.1.2 In the TIDS state the END and RQS messages shall be sent passive false and the NUL messages shall be sent passive true.

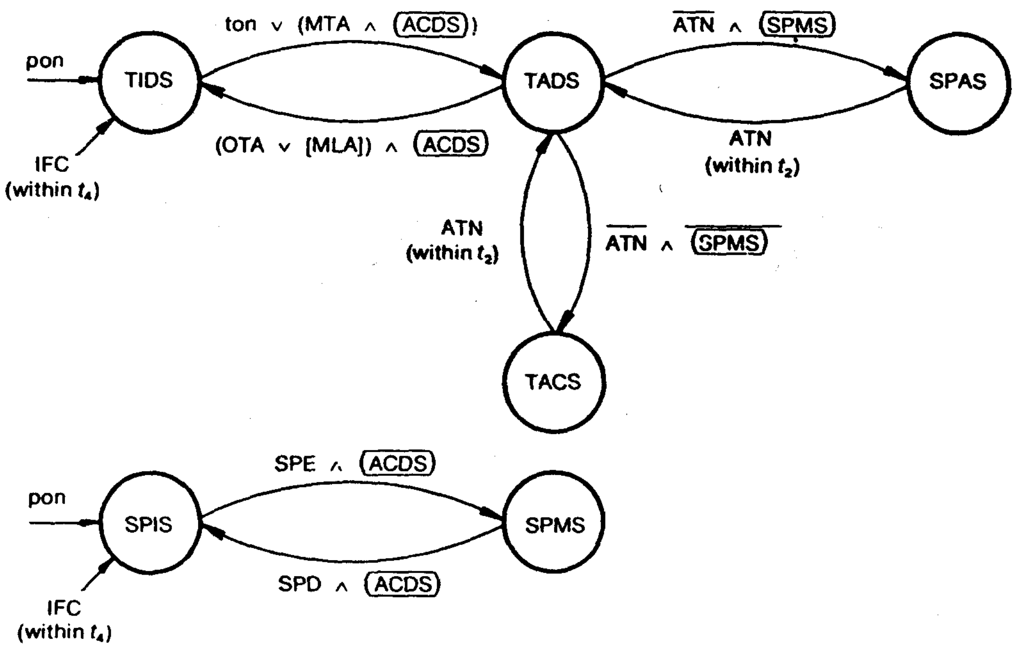


Figure 5 – Talker state diagram

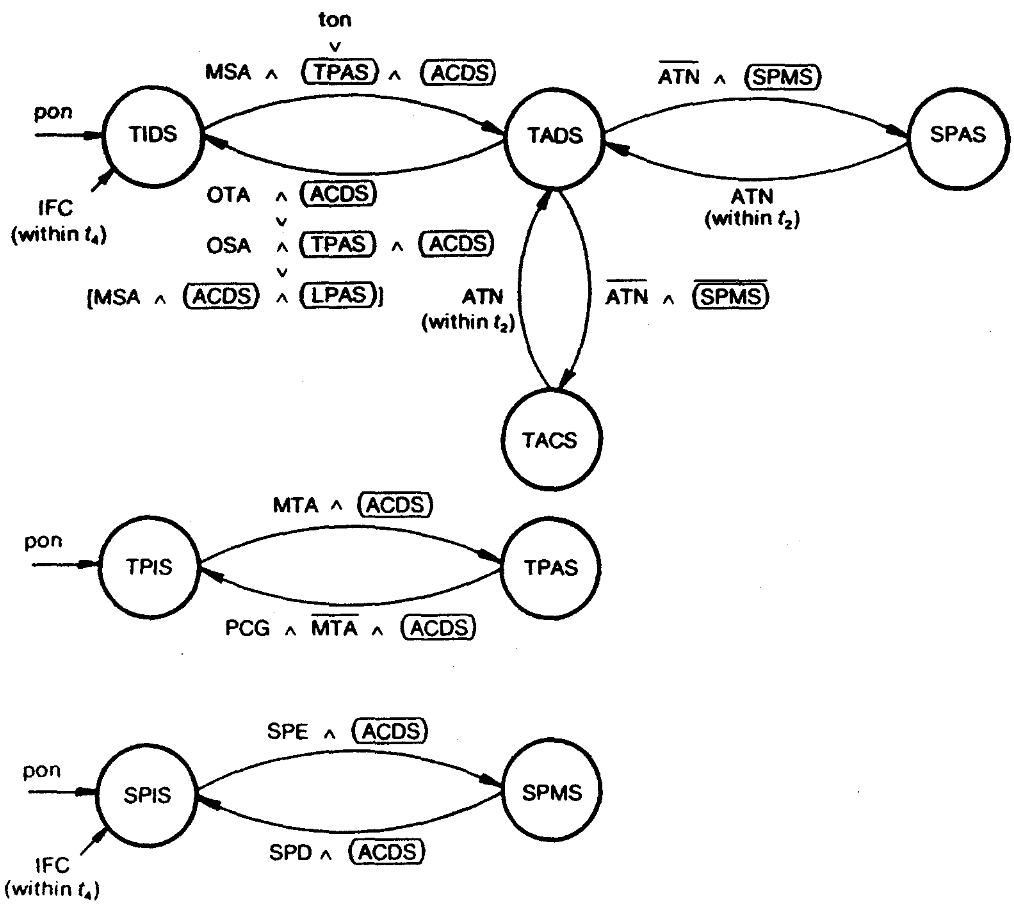
TABLE VIII – T mnemonics

Messages:		**Interface states:**	
pon	– power on	TIDS	– TALKER IDLE STATE
ton	– talk only	TADS	– TALKER ADDRESSED STATE
IFC	– INTERFACE CLEAR	TACS	– TALKER ACTIVE STATE
ATN	– ATTENTION	SPAS	– SERIAL POLL ACTIVE STATE
MTA	– MY TALK ADDRESS	SPIS	– SERIAL POLL IDLE STATE
SPE	– SERIAL POLL ENABLE	SPMS	– SERIAL POLL MODE STATE
SPD	– SERIAL POLL DISABLE		
OTA	– OTHER TALK ADDRESS	ACDS	– ACCEPT DATA STATE (AH FUNCTION)
MLA	– MY LISTEN ADDRESS		

TABLE IX – T or TE message outputs

T state	Qualifier	Remote message sent[2]			Device function interaction (DF)
		Multiligne	END	RQS[4]	
TIDS		(NUL)	(F)	(F)	Device function not allowed to send messages
TADS		(NUL)	(F)	(F)	Device function not allowed to send messages
TACS		DAB[1] or EOS[1]	T or F[1]	(F)	Device function can send DAB, EOS or END message(s) concurrent with DAB[3]
SPAS	APRS[5] inactive	STB[1]	F or T	F	Device function can send one STB message[3]
SPAS	APRS[5] active	STB[1]	F or T	T	Device function can send one STB message[3]

1) Messages enabled by the T function originating within the device functions.
2) See table XXXVIII (clause 16).
3) Under SH control.
4) See 8.3.4.1.
5) See 10.3.3.



NOTE - If the TE function is used together with the L function then [MSA ∧ ACDS ∧ LPAS] shall be replaced by [MLA ∧ ACDS].

Figure 6 – Extended talker state diagram

TABLE X – TE mnemonics

Messages:		**Interface states:**	
pon	– power on	TIDS	– TALKER IDLE STATE
ton	– talk only	TADS	– TALKER ADDRESSED STATE
IFC	– INTERFACE CLEAR	TACS	– TALKER ACTIVE STATE
ATN	– ATTENTION	SPAS	– SERIAL POLL ACTIVE STATE
MTA	– MY TALK ADDRESS	TPIS	– TALKER PRIMARY IDLE STATE
OTA	– OTHER TALK ADDRESS	TPAS	– TALKER PRIMARY ADDRESSED STATE
OSA	– OTHER SECONDARY ADDRESS	SPIS	– SERIAL POLL IDLE STATE
PCG	– PRIMARY COMMAND GROUP	SPMS	– SERIAL POLL MODE STATE
SPE	– SERIAL POLL ENABLE		
SPD	– SERIAL POLL DISABLE	ACDS	– ACCEPT DATA STATE (AH FUNCTION)
MSA	– MY SECONDARY ADDRESS	LPAS	– LISTENER PRIMARY ADDRESSED STATE (L FUNCTION)
T or TE message outputs (see table IX).			

8.3.1.3 The T function shall exit the TIDS state and enter the TALKER ADDRESSED STATE (TADS) if either:

– the MY TALK ADDRESS (MTA) message is true and the ACDS state is active;

– or the *talk only (ton)* message is true (see 8.5.3).

8.3.1.4 The TE function shall exit the TIDS state and enter the TADS state if either:

– the MY SECONDARY ADDRESS (MSA) message is true, and the ACDS state is active, and the TALKER PRIMARY ADDRESS STATE (TPAS) is active;

– or the *ton* message is true.

8.3.2 *TALKER ADDRESSED STATE (TADS)*

8.3.2.1 In the TADS state the T function has received its talk address and is prepared for, but not engaged in, sending data or status bytes. In the TADS state the TE function has received both its primary and secondary talk addresses and is prepared for, but not engaged in, sending data or status bytes.

8.3.2.2 In the TADS state the END and RQS messages shall be sent passive false and the NUL message shall be sent passive true.

8.3.2.3 The T function shall exit the TADS state and enter:

a) the TALKER ACTIVE STATE (TACS) if the ATN message is false and the SERIAL POLL MODE STATE (SPMS) is inactive;

b) the SERIAL POLL ACTIVE STATE (SPAS) if the ATN message is false and the SPMS state is active;

c) the TIDS state if either:

– the OTHER TALK ADDRESS (OTA) message is true and the ACDS state is active;

– or the MLA message is true and the ACDS state is active;

– or within t_4 if the IFC message is true.

NOTE - Use of the expression containing the MLA message is optional.

8.3.2.4 The TE function shall exit the TADS state and enter:

a) the TACS state if the ATN message is false and the SPMS state is inactive;

b) the SPAS state if the ATN message is false and the SPMS state is active;

c) the TIDS state if either:

- the OTA message is true and the ACDS state is active;
- or the OTHER SECONDARY ADDRESS (OSA) message is true and the TPAS and ACDS states are active;
- or the MSA message is true and both the LISTENER PRIMARY ADDRESSED STATE (LPAS) and ACDS states are active;
- or within t_4 if the INTERFACE CLEAR (IFC) message is true.

NOTE - Use of the expression containing the MSA message is optional. See also note under figure 6.

8.3.3 *TALKER ACTIVE STATE (TACS)*

8.3.3.1 In the TACS state the T function, or the TE function, enables the transfer of the DAB message and END, if used, from the device function to the interface signal lines. The message content is determined solely by the device function(s). The SH function determines when the device function(s) may change the message content of DAB (and END if used).

8.3.3.2 During the TACS state the DATA BYTE (DAB) or END OF STRING (EOS) and END messages may be sent *by the device functions*. The RQS message shall be sent passive false.

NOTE - The coding and format of the data are, in general, device-dependent and beyond the scope of this part.

8.3.3.3 The T function or the TE function shall exit the TACS state and enter:

a) the TADS state within t_2 if the ATN message is true;

b) the TIDS state within t_4 if the IFC message is true.

8.3.4 *SERIAL POLL ACTIVE STATE (SPAS)*

8.3.4.1 In the SPAS state the T function or the TE function enables the transfer of a single-status message from the device function to the interface signal lines using the SH function to control the transfer of the status byte that contains both the RQS and STB messages.

Although a controller needs only one byte for the STB and RQS messages from a device, it is allowable for the device to repeat this combined message byte if the controller does not assert ATN after the first transfer. In this case the content of STB message may change between subsequent transfers although the RQS message is held unaltered by the SR function.

8.3.4.2 During the SPAS state, whether the APRS state is active or inactive, the END message shall be sent either true or false. The RQS message shall be sent true if the APRS state is active, or false if the APRS state is inactive. In addition, the STB message shall be sent by the device function(s).

NOTE - The APRS state is contained in the Service Request Interface Function.

8.3.4.3 The T function or the TE function shall exit the SPAS state and enter:

a) the TADS state within t_2 if the ATN message is true;
b) the TIDS state within t_4 if the IFC message is true.

8.3.5 *SERIAL POLL IDLE STATE (SPIS)*

8.3.5.1 In the SPIS state the T function or the TE function is not enabled to participate in a serial poll. The T or TE function powers on in the SPIS state.

8.3.5.2 The SPIS state does not provide a remote message sending capability.

8.3.5.3 The T function or the TE function shall exit the SPIS state and enter the SPMS state if the SERIAL POLL ENABLE (SPE) message is true and the ACDS state is active.

8.3.6 *SERIAL POLL MODE STATE (SPMS)*

8.3.6.1 In the SPMS state the T function or the TE function is enabled to participate in a serial poll.

8.3.6.2 The SPMS state does not provide a remote message-sending capability.

8.3.6.3 The T function or the TE function shall exit the SPMS state and enter the SPIS state if either:

- the SERIAL POLL DISABLE (SPD) message is true and the ACDS state is active;

- or within t_4 if the IFC message is true.

8.3.7 *TALKER PRIMARY IDLE STATE (TPIS)*

8.3.7.1 In the TPIS state the TE function is able to recognize its primary address but not able to respond to its secondary address. The TE function powers on in the TPIS state.

8.3.7.2 The TPIS state does not provide a remote message-sending capability.

8.3.7.3 The TE function shall exit the TPIS state and enter the TPAS state if the MTA message is true and the ACDS state is active.

8.3.8 *TALKER PRIMARY ADDRESSED STATE (TPAS)*

8.3.8.1 In the TPAS state the TE function is able to recognize and respond to its secondary address.

8.3.8.2 The TPAS state does not provide a remote message-sending capability.

8.3.8.3 The TE function shall exit the TPAS state and enter the TPIS state if the PRIMARY COMMAND GROUP (PCG) message is true, and the MTA message is false, and the ACDS state is active.

8.4 *Talker function and extended talker function allowable subsets*

The only allowable subsets to the talker and extended talker interface functions shall be those listed in tables XI and XII.

TABLE XI

Identification	Description				States omitted	Other requirements	Other function subsets required
	Capabilities:						
	Basic talker	Serial poll	Talk only mode	Unaddress if MLA			
TØ	N	N	N	N	All	None	None
T1	Y	Y	Y	N	None	Omit $[MLA \wedge \boxed{ACDS}]$	SH1 and AH1
T2	Y	Y	N	N	None	1. Omit $[MLA \wedge \boxed{ACDS}]$ 2. ton always false	SH1 and AH1
T3	Y	N	Y	N	SPIS, SPMS and SPAS	1. Omit $[MLA \wedge \boxed{ACDS}]$	SH1 and AH1
T4	Y	N	N	N	SPIS, SPMS and SPAS	1. Omit $[MLA \wedge \boxed{ACDS}]$ 2. ton aways false	SH1 and AH1
T5	Y	Y	Y	Y	None	Include $[MLA \wedge \boxed{ACDS}]$	SH1 and L1-L4 or LE1-LE4
T6	Y	Y	N	Y	None	1. Include $[MLA \wedge \boxed{ACDS}]$ 2. ton alway false	SH1 and L1-L4 or LE1-LE4
T7	Y	N	Y	Y	SPIS, SPMS and SPAS	Include $[MLA \wedge \boxed{ACDS}]$	SH1 and L1-L4 or LE1-LE4
T8	Y	N	N	Y	SPIS, SPMS and SPAS	1. Include $[MLA \wedge \boxed{ACDS}]$ 2. ton always false	SH1 and L1-L4 or LE1-LE4

TABLE XII

Identification	Description					States omitted	Other requirements	Other function subsets required
	Capabilities:							
	Basic extended talker	Serial poll	Talk only mode	Unaddress if MSA ∧ LPAS				
TEØ	N	N	N	N		All	None	None
TE1	Y	Y	Y	N		None	Omit [MSA ∧ LPAS ∧ ACDS]	SH1 and AH1
TE2	Y	Y	N	N		None	1. Omit [MSA ∧ LPAS ∧ ACDS] 2. ton always false	SH1 and AH1
TE3	Y	N	Y	N		SPIS, SPMS and SPAS	Omit [MSA ∧ LPAS ∧ ACDS]	SH1 and AH1
TE4	Y	N	N	N		SPIS, SPMS and SPAS	1. Omit [MSA ∧ LPAS ∧ ACDS] 2. ton always false	SH1 and AH1
TE5	Y	Y	Y	Y		None	Include [MSA ∧ LPAS ∧ ACDS]	SH1 and L1-L4 or LE1-LE4
TE6	Y	Y	N	Y		None	1. Include [MSA ∧ LPAS ∧ ACDS] 2. ton always false	SH1 and L1-L4 or LE1-LE4
TE7	Y	N	Y	Y		SPIS, SPMS and SPAS	Include [MSA ∧ LPAS ∧ ACDS]	SH1 and L1-L 4 or LE1-LE4
TE8	Y	N	N	Y		SPIS, SPMS and SPAS	1. Include [MSA ∧ LPAS ∧ ACDS] 2. ton always false	SH1 and L1-L4 or LE1-LE4

8.5 *Additional T and TE interface function requirements and guidelines*

8.5.1 Each device which includes a T function or a TE function shall provide a means by which the talk address (or secondary address) which it recognizes as MTA (or MSA) can be changed in the field by the user of the device.

8.5.2 The interruption of a device sending data by transitions in and out of the TACS state should not adversely affect the format of the output data. It is recommended that a device returning to TACS state should continue with the output data string at the point of interruption.

8.5.3 Each device that includes the *ton* message shall be provided with a local means to generate the TALK ONLY function. It is intended that the *ton* message be used in a bus system with no C interface function capability.

9 Listener interface (L) function

9.1 *General description*

9.1.1 The L function provides a device with the capability to receive device-dependent data (including status data) over the interface from other devices. This capability exists only when the L function is addressed to listen.

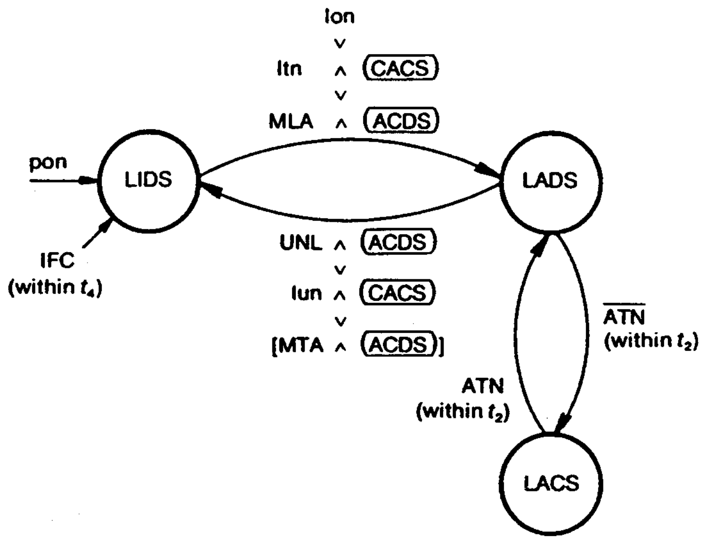


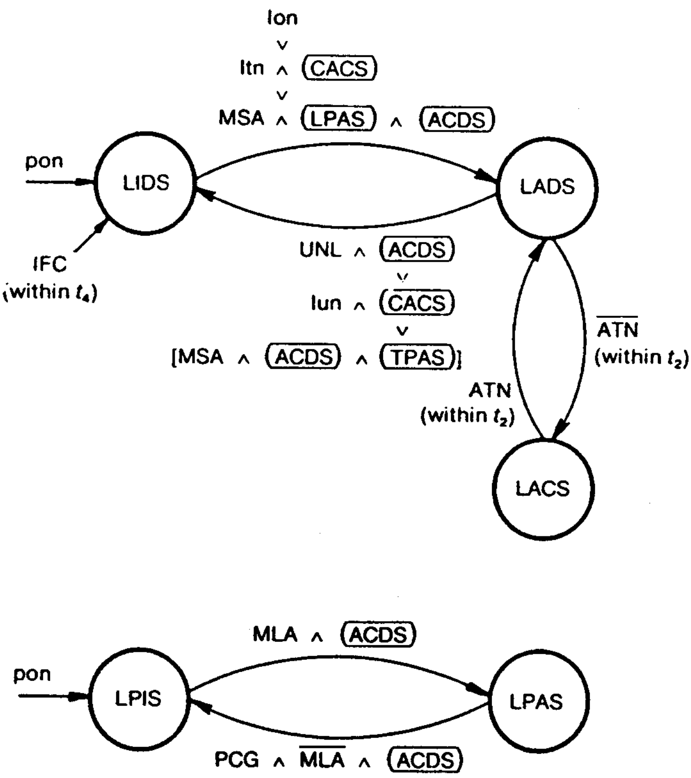
Figure 7 – Listener state diagram

TABLE XIII – L mnemonics

Messages:		**Interface states:**	
pon	– power on	LIDS	– LISTENER IDLE STATE
ltn	– listen	LADS	– LISTENER ADDRESSED STATE
lun	– local unlisten	LACS	– LISTENER ACTIVE STATE
lon	– listen only	ACDS	– ACCEPT DATA STATE (AH FUNCTION)
IFC	– INTERFACE CLEAR	CACS	– CONTROLLER ACTIVE STATE (C FUNCTION)
ATN	– ATTENTION		
UNL	– UNLISTEN		
MLA	– MY LISTEN ADDRESS		
MTA	– MY TALK ADDRESS		

TABLE XIV – L or LE message outputs

L or LE state	Remote message sent	Device function interaction
LIDS	None	Device function not allowed to receive messages
LADS	None	Device function not allowed to receive messages
LACS	None	Device function can receive one device-dependent message byte each time ACDS state is active



NOTE - If the LE function is used together with the T function then [MSA ∧ ACDS ∧ TPAS] shall be replaced by [MTA ∧ ACDS].

Figure 8 – Extended listener state diagram

TABLE XV – LE mnemonics

Messages:		**Interface states:**	
pon	– power on	LIDS	– LISTENER IDLE STATE
ltn	– listen	LACS	– LISTENER ACTIVE STATE
lun	– local unlisten	LADS	– LISTENER ADDRESSED STATE
lon	– listen only	LPIS	– LISTENER PRIMARY IDLE STATE
IFC	– INTERFACE CLEAR	LPAS	– LISTENER PRIMARY ADDRESSED STATE
ATN	– ATTENTION	ACDS	– ACCEPT DATA STATE (AH FUNCTION)
UNL	– UNLISTEN	CACS	– CONTROLLER ACTIVE STATE (C FUNCTION)
MLA	– MY LISTEN ADDRESS	TPAS	– TALKER PRIMARY ADDRESSED STATE (T FUNCTION)
PCG	– PRIMARY COMMAND GROUP		
MSA	– MY SECONDARY ADDRESS		
L or LE message outputs (see table XIV).			

9.1.2 There are two alternative versions of the function: one with and one without address extension. The normal L function uses a 1-byte address, the primary listen address. The L function with address extension (hereinafter called an LE (extended listener) function) uses a 2-byte address, the primary and secondary listen addresses. In all other respects, the capabilities of both versions are the same.

9.1.3 Only one of the two alternative L functions need be implemented in a specific device.

NOTE - Both the L function and LE function are described concurrently throughout clause 9 due to the extensive similarity between these two functions.

9.2 *Listener function state diagram*

9.2.1 The L function shall be implemented so as to perform according to the state diagram given in figure 7 and the state descriptions given throughout clause 9. Table XIII specifies the set of messages and states required to effect transition from one active state to another. Table XIV describes the device function interaction required while each state is active.

9.2.2 The LE function shall be implemented so as to perform according to the state diagram in figure 8 and the state descriptions given throughout clause 9. Table XV specifies the set of messages and states required to effect transition from one active state to another. Table XIV describes the device function interaction required while each state is active.

9.3 *Listener function state descriptions*

9.3.1 *LISTENER IDLE STATE (LIDS)*

9.3.1.1 In the LIDS state either the L function or the LE function is not engaged in the transfer of device-dependent messages. The L or LE function powers on in the LIDS state.

9.3.1.2 The LIDS state does not provide a remote message-sending capability.

9.3.1.3 The L function shall exit the LIDS state and enter the LISTENER ADDRESSED STATE (LADS) if:

- the MY LISTEN ADDRESS (MLA) message is true and the ACDS state is active;
- or the *listen only (lon)* message is true (see 9.5.3);
- or the *listen (ltn)* message is true and the CACS state is active.

9.3.1.4 The LE function shall exit the LIDS state and enter the LADS state if either:

- the MY SECONDARY ADDRESS (MSA) message is true and the ACDS state is active, and the LISTENER PRIMARY ADDRESSED STATE (LPAS) is active;
- or the *lon* message is true;
- or the *ltn* message is true and the CACS state is active.

9.3.2 *LISTENER ADDRESSED STATE (LADS)*

9.3.2.1 In the LADS state the L function has received its listen address and is prepared for, but not engaged in, the transfer of device-dependent messages. In the LADS state the LE function has received both its primary and secondary listen addresses and is prepared for, but not engaged in, the transfer of device-dependent messages.

9.3.2.2 The LADS state does not provide a remote message-sending capability.

9.3.2.3 The L function shall exit the LADS state and enter:

a) the LISTENER ACTIVE STATE (LACS) within t_2 if the ATN message is false;

b) the LIDS state if either:
   - the UNLISTEN (UNL) message is true and the ACDS state is active;
   - or the *local unlisten (lun)* message is true and the CACS state is active;
   - or the MTA message is true and the ACDS state is active;
   - or within t_4 if the IFC message is true.

NOTE - Use of the expression containing the MTA message is optional.

9.3.2.4 The LE function shall exit the LADS state and enter:

a) the LACS state within t_2 if the ATN message is false;
b) the LIDS state if either:
   - the UNL message is true and the ACDS state is active;
   - or the *lun* message is true and the CACS state is active;
   - or the MSA message is true and the TPAS and ACDS states are active;
   - or within t_4 if the IFC message is true.

NOTE - Use of the expression containing the MSA message is optional. See also note under figure 8.

9.3.3 *LISTENER ACTIVE STATE (LACS)*

9.3.3.1 In the LACS state the L function, or the LE function, is enabled to transfer any device-dependent message (DAB, EOS, STB, END or RQS) to the device functions as received via the interface signal lines. The AH function is used by the device function(s) to control the message transfer.

NOTE - The coding and format of the data are, in general, device-dependent and beyond the scope of this part.

9.3.3.2 The LACS state does not provide a remote message-sending capability.

9.3.3.3 The L function or the LE function shall exit the LACS state and enter:

a) the LADS state within t_2 if the ATN message is true;
b) the LIDS state within t_4 if the IFC message is true.

9.3.4 *LISTENER PRIMARY IDLE STATE (LPIS)*

9.3.4.1 In the LPIS state the LE function is able to recognize its primary address but is not able to respond to its secondary address.

The LE function powers on in the LPIS state.

9.3.4.2 The LPIS state does not provide a remote message-sending capability.

9.3.4.3 The LE function shall exit the LPIS state and enter the LPAS state if the MLA message is true, and the ACDS state is active.

9.3.5 *LISTENER PRIMARY ADDRESSED STATE (LPAS)*

9.3.5.1 In the LPAS state the LE function is able to recognize and respond to its secondary address.

9.3.5.2 The LPAS state does not provide a remote message-sending capability.

9.3.5.3 The LE function shall exit the LPAS state and enter the LPIS state if the PRIMARY COMMAND GROUP (PCG) message is true, the MLA message is false, and the ACDS state is active.

9.4 *Listener function and extended listener function allowable subsets*

The only allowable subsets to the L and LE functions shall be those listed in tables XVI and XVII.

TABLE XVI

Identi-fication	Description				States omitted	Other requirements	Other function subsets required
	Capabilities:						
	Basic listener	Listen only mode	Unaddress if MTA				
LØ	N	N	N		All	None	None
L1	Y	Y	N		None	Omit [MTA ∧ (ACDS)]	AH1
L2	Y	N	N		None	1. Omit [MTA ∧ (ACDS)] 2. lon always false	AH1
L3	Y	Y	Y		None	Include [MTA ∧ (ACDS)]	AH1 and T1-T8 or TE1-TE8
L4	Y	N	Y		None	1. Include [MTA ∧ (ACDS)] 2. lon always false	AH1 and T1-T8 or TE1-TE8

TABLE XVII

Identification	Description				States omitted	Other requirements	Other function subsets required
	Capabilities:						
	Basic extended listener	Listen only mode	Unaddress if [MSA ∧ TPAS]*				
LE0	N	N	N		All	None	None
LE1	Y	Y	N		None	Omit [MSA ∧ TPAS ∧ ACDS]	AH1
LE2	Y	N	N		None	1. Omit [MSA ∧ TPAS ∧ ACDS] 2. lon always false	AH1
LE3	Y	Y	Y		None	Include [MSA ∧ TPAS ∧ ACDS]	AH1 and T1-T8 or TE1-TE8
LE4	Y	N	Y		None	1. Include [MSA ∧ TPAS ∧ ACDS] 2. lon always false	AH1 and T1-T8 or TE1-TE8

* Replaced by MTA when used together with the T function.

9.5 *Additional L or LE requirements and guidelines*

9.5.1 Each device that includes an L function (or LE function) shall provide a means by which the listen address (or secondary address) that it recognizes as MLA (or MSA) can be changed.

9.5.2 The interruption of a device receiving data by transitions in and out of the LACS state should not adversely affect the future receipt of input data. It is recommended that a device returning to the LACS state should continue with the input data string at the point of interruption.

9.5.3 Each device that includes the *lon* message shall be provided with a local means to generate the listen only condition. It is intended that the *lon* message be used in a bus system with no C interface function capability.

10 Service request interface (SR) function

10.1 *General description*

The SR function provides a device with the capability to request service asynchronously from the controller in charge of the interface.

It also synchronizes the content of the RQS message of the composite status byte present during a serial poll so that the SRQ message can be removed from the interface once the RQS message is received true by the controller in charge (see 15.1).

10.2 *Service Request interface function state diagram*

The SR function shall be implemented so as to perform according to the state diagram given in figure 9 and the state descriptions given throughout clause 10. Table XVIII specifies the set of messages and states required to effect transition from one active state to another. Table XIX specifies the messages that shall be sent and the device function interaction required while each state is active.

10.3 *Service Request state description*

10.3.1 *NEGATIVE POLL RESPONSE STATE (NPRS)*

10.3.1.1 In the NPRS state the SR function is not requesting service.

The SR function powers on in the NPRS state.

10.3.1.2 In the NPRS state the SRQ message shall be sent passive false.

NOTE - The RQS message will be sent false when the SPAS state is active (see 8.3.4).

10.3.1.3 The SR function shall exit the NPRS state and enter the SRQS state at any time the *request service (rsv)* message is true and the SPAS state is not active.

10.3.2 *SERVICE REQUEST STATE (SRQS)*

10.3.2.1 In the SRQS state the SR function continuously indicates over the interface that it is requesting service.

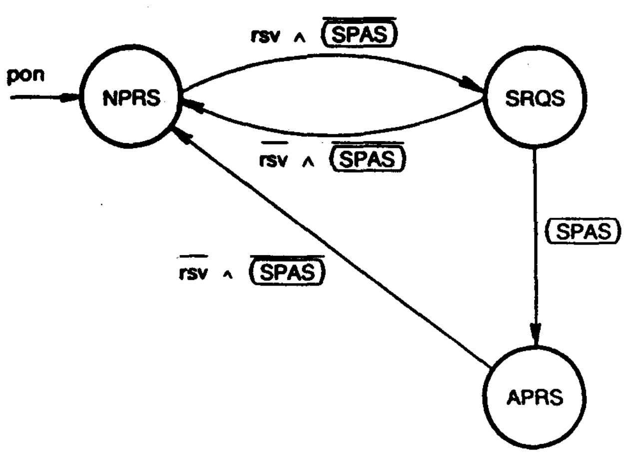


Figure 9 – Service request state diagram

TABLE XVIII – SR mnemonics

Messages:	**Interface states:**
pon – power on rsv – request service	NPRS – NEGATIVE POLL RESPONSE STATE SRQS – SERVICE REQUEST STATE APRS – AFFIRMATIVE POLL RESPONSE STATE (SPAS) – SERIAL POLL ACTIVE STATE (T FUNCTION)

TABLE XIX – SR message outputs

SR state	Remote message sent	Device function interaction (DF)
	SRQ	
NPRS	(F)	None
SRQS	T	
APRS	(F)	

10.3.2.2 In the SRQS state the SRQ message shall be sent true.

10.3.2.3 The SR function shall exit the SRQS state and enter:

a) the NPRS state if the *rsv* message is false and the SPAS state is not active;

b) the AFFIRMATIVE POLL RESPONSE STATE (APRS) if the SPAS state is active.

10.3.3 *AFFIRMATIVE POLL RESPONSE STATE (APRS)*

10.3.3.1 In the APRS state the SR function requires service but does not actively request it over the interface.

10.3.3.2 In the APRS state the SRQ message shall be sent passive false.

NOTE - The RQS message will be sent true by the talker when the SPAS state is active (see 8.3.4).

10.3.3.3 The SR function shall exit the APRS state and enter the NPRS state at any time the *rsv* message is false and the SPAS state is not active.

10.4 *Service Request interface function allowable subsets*

The only allowable subsets to the SR function shall be those listed in table XX.

TABLE XX

Identification	Description	States omitted	Other requirements	Other function subsets required
SR0	No capability	All	None	None
SR1	Complete capability	None	None	T1, T2, T5, T6, TE1, TE2, TE5 or TE6

10.5 *Additional SR interface function requirements and guidelines*

10.5.1 An SR function is required for each unique reason for requesting service. If more than one reason exists, within a device, then a separate SR function and corresponding *rsv* message shall be used for each separate reason.

10.5.1.1 Preferred practice is to have logical OR multiple conditions within a device to generate a single reason for requesting service for a single SR function. If multiple SR functions are used, a single SRQ true message should be sent when requested.

10.5.1.2 While the T function is in the SPAS state, the RQS message shall be sent true if any of the SR functions, within a device, is in the APRS state.

10.5.2 The SRQ message received, via the C function, is the logical OR of the SRQ messages sent by all SR functions. The way this is performed via the use of the SRQ signal line is explained in 33.2.

11 Remote/local interface (RL) function

11.1 *General description*

The RL interface function provides a device with the capability to enable and disable its local controls.

11.2 *Remote/local function state diagram*

The RL function shall be implemented so as to perform according to the state diagram given in figure 10 and the state descriptions given throughout clause 11. Table XXI specifies the set of messages and states required to effect transition from one active state to another. Table XXII specifies the device function interaction required while each state is active.

11.3 *Remote/local state descriptions*

11.3.1 *LOCAL STATE (LOCS)*

11.3.1.1 In the LOCS state all local controls of the associated device functions are operative and the device may respond to corresponding device-dependent messages from the interface. The RL function powers on in the LOCS state.

11.3.1.2 The LOCS state does not provide a remote message-sending capability.

11.3.1.3 The RL function shall exit the LOCS state and enter:

a) the REMOTE STATE (REMS) if the *return to local (rtl)* message is false and the MLA message is true and the ACDS state is active;

b) the LOCAL WITH LOCKOUT STATE (LWLS) if the universal coded command LOCAL LOCKOUT (LLO) is true and the ACDS state is active.

11.3.2 *LOCAL WITH LOCKOUT STATE (LWLS)*

11.3.2.1 In the LWLS state all local device controls of the associated device functions are operative and the device may respond to corresponding device-dependent messages from the interface. The *rtl* message is ignored.

11.3.2.2 The LWLS state does not provide a remote message-sending capability.

11.3.2.3 The RL function shall exit the LWLS state and enter:

a) the REMOTE WITH LOCKOUT STATE (RWLS) when MLA is true and the ACDS state is active;

b) the LOCS state within t_4 if the REN message is false.

11.3.3 *REMOTE STATE (REMS)*

11.3.3.1 In the REMS state some or all of the local controls (of the associated device functions) which have corresponding remote controls, except those controls which send local messages to interface functions, may be inoperative.

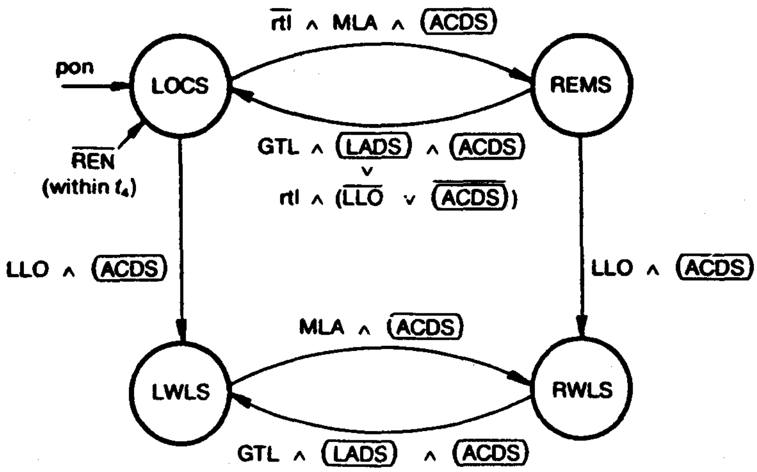
11.3.3.2 The REMS state does not provide a remote message-sending capability.

11.3.3.3 The RL function shall exit the REMS state and enter:

a) the RWLS state if the LLO message is true and the ACDS state is active;

b) the LOCS state:

- within t_4 if the REN message is false;
- or the GO TO LOCAL (GTL) message is true and the ACDS and LADS states are active;
- or the *rtl* message is true and either the LLO message is false or the ACDS state is inactive.



NOTE - If the RL function is used together with the LE function then the term MLA shall be replaced by the term MSA $\wedge$ (LPAS).

Figure 10 – Remote local state diagram

TABLE XXI – RL mnemonics

Messages:		Interface states:	
pon	– power on	LOCS	– LOCAL STATE
rtl	– return to local	LWLS	– LOCAL WITH LOCKOUT STATE
REN	– REMOTE ENABLE	REMS	– REMOTE STATE
LLO	– LOCAL LOCKOUT	RWLS	– REMOTE WITH LOCKOUT STATE
GTL	– GO TO LOCAL	ACDS	– ACCEPT DATA STATE (AH FUNCTION)
MLA	– MY LISTEN ADDRESS	LADS	– LISTENER ADDRESSED STATE (L FUNCTION)
MSA	– MY SECONDARY ADDRESS	LPAS	– LISTENER PRIMARY ADDRESSED STATE (LE FUNCTION)

TABLE XXII – RL message outputs

RL state	Remote message sent	Device function interaction (DF)
LOCS	None	Device is in "local control" mode
LWLS		Device is in "local control" mode
REMS		Device is in "remote control" mode
RWLS		Device is in "remote control" mode

11.3.4 *REMOTE WITH LOCKOUT STATE (RWLS)*

11.3.4.1 In the RWLS state some or all of the local controls (of the associated device functions) which have corresponding remote controls, except those controls which send local messages to interface functions, may be inoperative. The *rtl* message is ignored.

11.3.4.2 The RWLS state does not provide a remote message-sending capability.

11.3.4.3 The RL function shall exit the RWLS state and enter:

a) the LOCS state within t_4 if the REN message is false;

b) the LWLS state if the GTL message is true and the LADS and ACDS states are active.

11.4 *Remote/Local function allowable subsets*

The only allowable subsets to the RL function shall be those listed in table XXIII.

TABLE XXIII

Identification	Description	States omitted	Other requirements	Other function subsets required
RLØ	No capability	All	None	None
RL1	Complete capability	None	None	L1-L4 or LE1-LE4
RL2	No local lockout	LWLS and RWLS	*rtl* always false	L1-L4 or LE1-LE4

11.5 *Additional RL function requirements and guidelines*

11.5.1 The ability of a device either to send device-dependent messages over the interface or to receive and utilize device-dependent messages not in conflict with locally available data is independent of the state which is active within the RL function.

11.5.2 When either the REMS or RWLS state is active, the associated device shall become responsive to all subsequent input data received via the interface. Local controls shall be ignored unless specifically enabled by device-dependent messages sent after entering REMS or RWLS.

It is recommended that the device not alter its state (including local controls) as a result of a transition from LOCS to REMS or from LWLS to RWLS.

11.5.3 Conversely, when either the LOCS or LWLS states becomes active, the associated device shall become responsive to future use of local controls.

After a transition for REMS or RWLS to LOCS or LWLS, it is recommended that devices, whose indicators (mechanical, positional, etc.) cannot be changed by remote control, alter their local controls (and device state variables) as necessary for their front panel indicators and device state to agree.

After a transition from REMS or RWLS to LOCS or LWLS, it is recommended that devices, whose front panel indicators can be changed by remote control, alter their indicators as necessary for their front panel indication and device state to agree.

11.5.4 The *rtl* message shall not be generated permanently.

11.5.5 Applications that require absolute local control of a device by a local programming source (e.g. a human operator) are beyond the scope of this part.

12 Parallel poll interface (PP) function

12.1 *General description*

12.1.1 The PP function provides a device with the capability to present a PPR message to the controller-in-charge without being previously addressed to talk.

12.1.2 The signal lines DIO1 to DIO8 are used to convey the device status bits during the parallel poll. In order for a device to respond with a PPR message a device shall have been assigned (configured) to a single DIO line either by the controller or by a local message. This allows up to eight devices with a one-line-per-device assignment, although any number of devices can be handled through sharing of DIO lines.

12.1.3 The use of the parallel poll facility within a system requires a commitment of the current interface controller to conduct a parallel poll, as required.

12.1.4 The parallel poll facility can be used to indicate a request for service. This capability differs from use of the SRQ message in the following ways:

– A controller initiates a parallel poll sequence whereas any device requests the initiation of a serial poll sequence.

– A parallel poll enables the transfer of status data from multiple devices concurrently whereas a serial poll sequentially collects status data from each device.

12.2 *Parallel poll function state diagram*

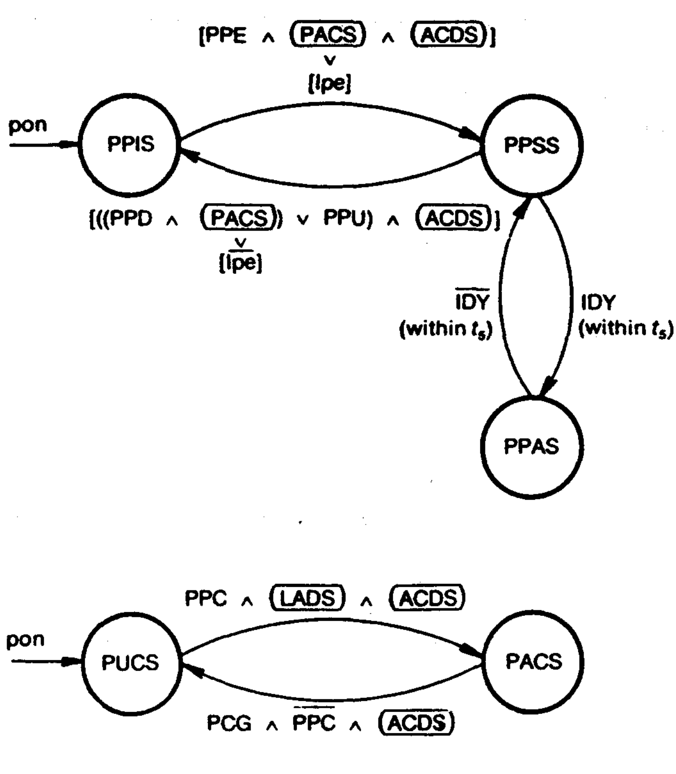
12.2.1 The PP function shall be implemented so as to perform according to the state diagram given in figure 11 and the state descriptions given throughout clause 12. Table XXIV specifies the set of messages and states required to effect transition from one active state to another. Table XXV specifies the messages that shall be sent and the device function interaction required by the function while each state is active.

12.3 *Parallel poll state descriptions*

12.3.1 *PARALLEL POLL IDLE STATE (PPIS)*

12.3.1.1 In the PPIS state the PP function is unable to respond to a parallel poll issued by the interface controller. The PP function powers on in the PPIS state.

12.3.1.2 In the PPIS state all PPR messages shall be sent passive false.



NOTE - See table XXVII for restrictions on use of the optional transitions.

Figure 11 – Parallel poll state diagram

TABLE XXIV – PP mnemonics

Messages:		Interface states:	
pon	– power on	PPIS	– PARALLEL POLL IDLE STATE
ist	– individual status (table XXV)		
lpe	– local poll enabled	PPSS	– PARALLEL POLL STANDBY STATE
IDY	– IDENTIFY	PPAS	– PARALLEL POLL ACTIVE STATE
PPE	– PARALLEL POLL ENABLE	PUCS	– PARALLEL POLL UNADDRESSED TO CONFIGURE STATE
PPD	– PARALLEL POLL DISABLE	PACS	– PARALLEL POLL ADDRESSED TO CONFIGURE STATE
PPC	– PARALLEL POLL CONFIGURE	ACDS	– ACCEPT DATA STATE (AH FUNCTION)
PCG	– PRIMARY COMMAND GROUP	LADS	– LISTENER ADDRESSED STATE (L FUNCTION)
PPU	– PARALLEL POLL UNCONFIGURE		

12.3.1.3 The PP function shall exit the PPIS state and enter the PARALLEL POLL STANDBY STATE (PPSS) for one and only one of the following expressions:

– the PPE message is true and the PACS and ACDS states are active;

– or the *local poll enabled (lpe)* message is true.

12.3.2 *PARALLEL POLL STANDBY STATE (PPSS)*

12.3.2.1 In the PPSS state the PP function is able to respond to parallel polls issued by the device controller whenever they occur.

TABLE XXV – PP message outputs

PP state	Qualifier	Remote message sent PPRn[2]	Device function interaction
PPIS		(F)	None
PPSS		(F)	
PPAS	ist ≡ S[1]	T	
PPAS	ist ≢ S[1]	(F)	

[1] See 12.3.3.2.

[2] This column refers only to the specific message assigned by the device.

12.3.2.2 In the PPSS state all PPR messages shall be sent passive false.

12.3.2.3 The PP function shall exit the PPSS state and enter:

a) the PARALLEL POLL ACTIVE STATE (PPAS) within t_5 if the IDY message is true (a parallel poll is in progress);

b) the PPIS state for one and only one of the following expressions:

- the *lpe* message is false;
- or the PARALLEL POLL DISABLE (PPD) message is true and the PACS and ACDS states are active, or the PARALLEL POLL UNCONFIGURE (PPU) message is true and the ACDS state is active.

12.3.3 *PARALLEL POLL ACTIVE STATE (PPAS)*

12.3.3.1 In the PPAS state the PP function is responding to the parallel poll currently being conducted by the interface controller.

12.3.3.2 In the PPAS state one of the PPR messages shall be sent true if, and only if, the value of the *individual status (ist)* message is equal to the value of the SENSE (S) bit received as part of the most recently received PPE command. The PPR message to be sent shall be the one specified by the three bits P1 to P3 received as part of the most recently received PPE command. Table XXVI lists the PPR message specified by each of the combinations of values of P1 to P3 (see 12.5.1). All other PPR messages should be sent passive false.

TABLE XXVI

Parallel poll message			
Bits received with most recent PPE command			PPR message specified
P3	P2	P1	
0	0	0	PPR1
0	0	1	PPR2
0	1	0	PPR3
0	1	1	PPR4
1	0	0	PPR5
1	0	1	PPR6
1	1	0	PPR7
1	1	1	PPR8

12.3.3.3 The parallel poll interface function shall exit the PPAS state and enter the PPSS state within t_5 if the IDY message is false (the parallel poll is finished).

12.3.4 *PARALLEL POLL UNADDRESSED TO CONFIGURE STATE (PUCS)*

12.3.4.1 In the PUCS state the PP function shall ignore any PPE or PPD messages which might be received over the interface. The PP function powers on in the PUCS state.

12.3.4.2 The PUCS state does not provide a remote message-sending capability.

12.3.4.3 The PP function shall exit the PUCS state and enter the PARALLEL POLL ADDRESSED TO CONFIGURE STATE (PACS) if the PPC message is true, and the LADS and ACDS states are active.

12.3.5 *PARALLEL POLL ADDRESSED TO CONFIGURE STATE (PACS)*

12.3.5.1 In the PACS state the PP function is able to act upon PPE or PPD messages received over the interface. If a PPE message is received, the attendant bits S, P1, P2 and P3 should be saved by the function.

12.3.5.2 The PACS state does not provide a remote message-sending capability.

12.3.5.3 The PP function shall exit the PACS state and enter the PARALLEL POLL UNADDRESSED TO CONFIGURE STATE (PUCS) when the PCG message is true, the PPC message is false, and the ACDS state is active.

12.4 *Parallel poll interface function allowable subsets*

The only allowable subsets to the parallel poll interface function shall be those listed in table XXVII.

TABLE XXVII

Identi-fication	Description	States omitted	Other requirements	Other function subsets required
PPØ	No capability	All	None	None
PP1	Remote configuration	None	1. Include [((PPD ∧ (PACS)) ∨ PPU) ∧ (ACDS)] 2. Include [PPE ∧ (PACS) ∧ (ACDS)] 3. Exclude *lpe*	L1-L4 or LE1-LE4
PP2	Local configuration	PUCS, PACS	1. Include *lpe* 2. Exclude [((PPD ∧ (PACS)) ∨ PPU) ∧ (ACDS)] 3. Exclude [PPE ∧ (PACS) ∧ (ACDS)] 4. Local messages shall be substituted for S, P1, P2, P3	None

12.5 *Additional PP interface function requirements and guidelines*

If subset PP2 is taken, field-settable local messages shall substitute for the PPE command to specify the PPR message and the message sense to be used during a parallel poll.

13 Device clear interface (DC) function

13.1 *General description*

The DC function provides the device with the capability to be cleared (initialized) either individually or as a part of a group of devices. The group may be either a subset or all addressed devices in one system.

13.2 *Device clear function state diagram*

The DC function shall be implemented so as to perform according to the state diagram given in figure 12 and the state descriptions given throughout clause 13. Table XXVIII specifies the set of messages and states required to effect transition from one active state to another. Table XXXIX specifies the device function interaction required while each state is active.

13.3 *Device clear function state descriptions*

13.3.1 *DEVICE CLEAR IDLE STATE (DCIS)*

13.3.1.1 In the DCIS state the DC function is inactive.

13.3.1.2 The DCIS state does not provide a remote message-sending capability.

13.3.1.3 The DC function shall exit the DCIS state and enter the DEVICE CLEAR ACTIVE STATE (DCAS) if the ACDS state is active, and either:

- the DEVICE CLEAR (DCL) message is true;
- or the SELECTED DEVICE CLEAR (SDC) message is true and the LADS state is active.

NOTE - Use of the expression containing the SDC message is optional.

13.3.2 *DEVICE CLEAR ACTIVE STATE (DCAS)*

13.3.2.1 In the DCAS state the DC function sends an internal message to the device function(s) causing it (them) to be cleared.

13.3.2.2 The DCAS state does not provide a remote message-sending capability.

13.3.2.3 The DC function shall exit the DCAS state and enter the DEVICE CLEAR IDLE STATE (DCIS) if either the ACDS state is inactive or neither:

- the DCL message is true;
- nor the SDC message is true and the LADS state is active.

NOTE - Use of the expression containing the SDC message is optional.

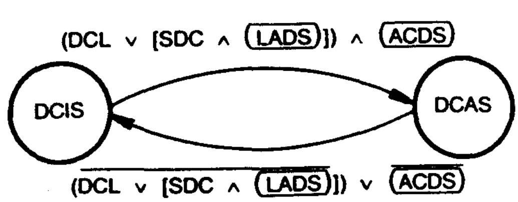


Figure 12 – Device clear state diagram

TABLE XXVIII – DC mnemonics

Messages:	Interface states:
DCL – DEVICE CLEAR SDC – SELECTED DEVICE CLEAR	DCIS – DEVICE CLEAR IDLE STATE DCAS – DEVICE CLEAR ACTIVE STATE ACDS – ACCEPT DATA STATE (AH FUNCTION) LADS – LISTENER ADDRESSED STATE (L FUNCTION)

TABLE XXIX – DC message outputs

DC state	Remote message sent	Device function interaction
DCIS DCAS	None	Normal device function operation Device function should return to a known fixed state

13.4 *Device clear interface function allowable subsets*

The only allowable subsets of the device clear interface function shall be those listed in table XXX.

TABLE XXX

Identification	Description	States omitted	Other requirements	Other function subsets required
DCØ	No capability	All	None	None
DC1	Complete capability	None	None	L1-L4 or LE1-LE4
DC2	Omit selective device clear	None	Omit [SDC $\wedge$ LADS]	AH1

13.5 *Additional DC function requirements and guidelines*

13.5.1 *Interpretation of the DCAS state*

13.5.1.1 The DCAS state affects only device functions and does not affect interface functions (cleared by IFC).

13.5.1.2 A device may use the DC function for any purpose consistent with its operation. Normally, use of the DC function should allow resumption of device dependent message flow to and from device functions. However, this function may be used to put any subset of the device's functions to a defined state deemed appropriate by the designer, which state the designer shall then specify.

14 Device trigger interface (DT) function

14.1 *General description*

The DT function provides the device with the capability to have its designated operation started either individually or as part of a group of devices. The group may be either a subset or all addressed devices in one system.

14.2 *Device trigger function state diagram*

The DT function shall be implemented so as to perform according to the state diagram given in figure 13 and the state descriptions given throughout clause 14. Table XXXI specifies the set of messages and states required to effect transition from one active state to another. Table XXXII specifies the device function interaction required while each state is active.

14.3 *Device trigger function state descriptions*

14.3.1 *DEVICE TRIGGER IDLE STATE (DTIS)*

14.3.1.1 In the DTIS state the DT function is inactive.

14.3.1.2 The DTIS state does not provide a remote message-sending capability.

14.3.1.3 The DT function shall exit the DTIS state and enter the DEVICE TRIGGER ACTIVE STATE (DTAS) if:

- the GROUP EXECUTE TRIGGER (GET) message is true;
- and the LADS and ACDS states are active.

14.3.2 *DEVICE TRIGGER ACTIVE STATE (DTAS)*

14.3.2.1 In the DTAS state the DT function sends an internal message to the device function causing it to start performing its basic operation.

14.3.2.2 The DTAS state does not provide a remote message-sending capability.

14.3.2.3 The DT function shall exit the DTAS state and enter the DTIS state if either:

- the GET message is false;
- or the LADS state is inactive;
- or the ACDS state is inactive.

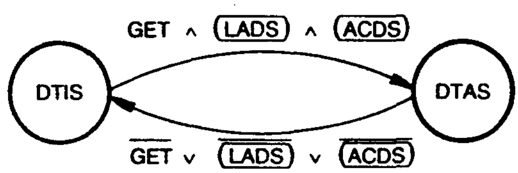


Figure 13 – Device trigger state diagram

TABLE XXXI – DT mnemonics

Messages:	**Interface states:**
GET – GROUP EXECUTE TRIGGER	DTIS – DEVICE TRIGGER IDLE STATE DTAS – DEVICE TRIGGER ACTIVE STATE (ACDS) – ACCEPT DATA STATE (AH FUNCTION) (LADS) – LISTENER ADDRESSED STATE (L FUNCTION)

TABLE XXXII – DT message outputs

DT state	Remote message sent	Device function interaction
DTIS	None	Normal device function operation
DTAS	None	Device function should start performing triggered operation

14.4 *Device trigger interface allowable subsets*

The only allowable subsets to the DT function shall be those listed in table XXXIII.

TABLE XXXIII

Identification	Description	States omitted	Other requirements	Other function subsets requested
DTØ	No capability	All	None	None
DT1	Complete capability	None	None	L1-L4 or LE1-LE4

14.5 *Additional DT function requirements and guidelines*

14.5.1 *Interpretation of the DTAS state*

14.5.1.1 The DTAS state indicates that the device (or defined portions of the device) is to start performing its designated operation.

14.5.1.2 Once a device operation has been started, it must not respond to subsequent state transitions until the operation is complete. Only after completion of the first operation can the device start a new operation in response to the next DTAS active condition.

15 Controller interface (C) function

15.1 *General description*

15.1.1 The C function provides a device with the capability to send device addresses, universal commands and addressed commands to other devices over the interface. It also provides the possibility of conducting parallel polls to determine which devices require service.

15.1.2 A C function can exercise its capabilities only when it is sending the ATN message over the interface.

15.1.3 If more than one device on the interface system has a C function, then all but one of them shall be in the CONTROLLER IDLE STATE (CIDS) at any given time. The device containing the C function which is not in the CIDS state is called the *controller-in-charge* (of the interface system). Protocol is provided within this part to allow devices with a C function to take turns as the controller-in-charge of the interface.

15.1.4 The C function in one of the devices connected to an interface (but no more than one) can exist in the SYSTEM CONTROL ACTIVE STATE (SACS). It shall remain in this state throughout operation of the interface and so possesses the capability to send the IFC and REN messages over the interface at any time whether or not it is the controller-in-charge. This device is called the *system controller* (of the interface system).

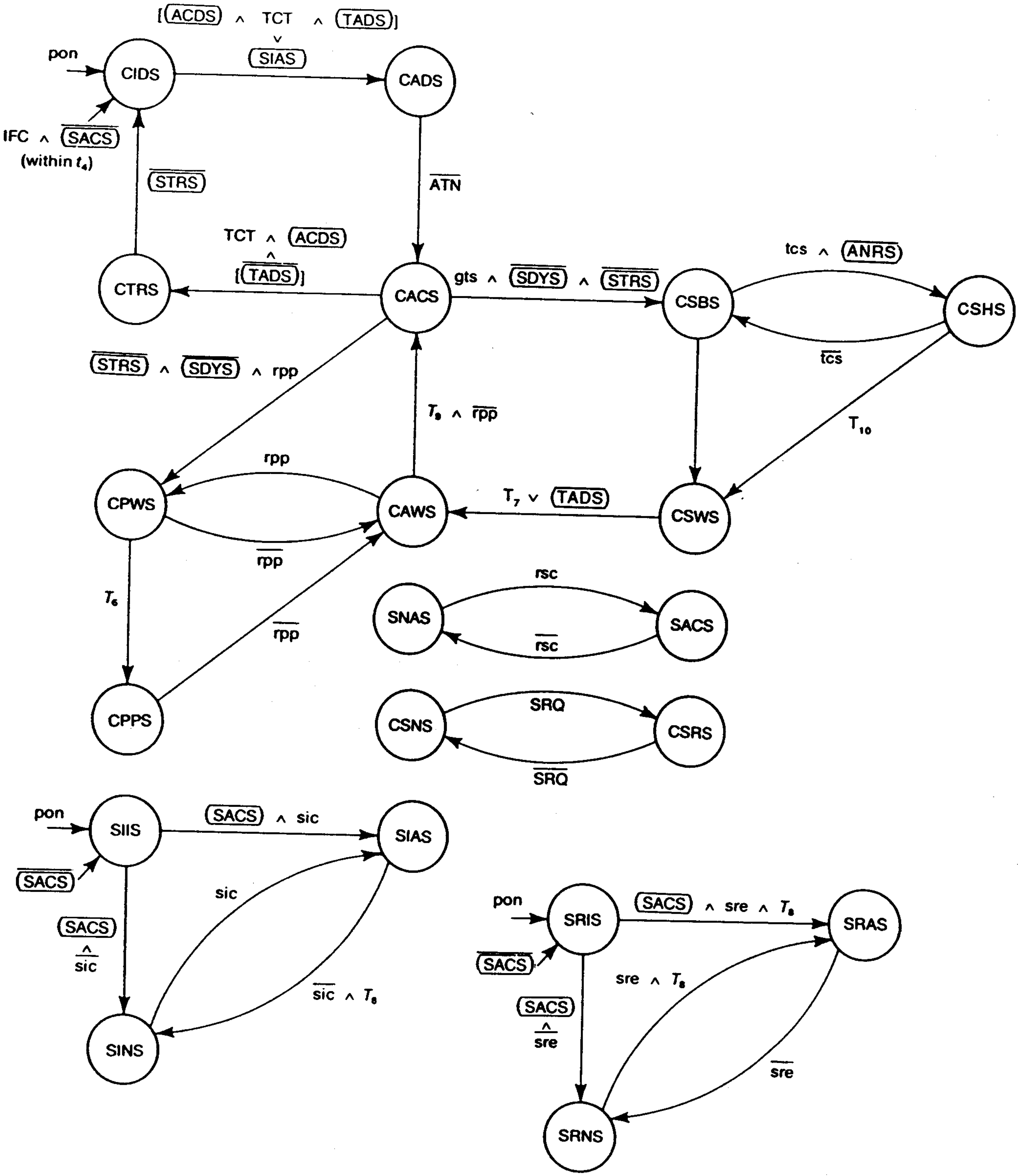


Figure 14 – Controller state diagram

15.2 *Controller function state diagram*

The C function shall be implemented so as to perform according to the state diagram given in figure 14 and the state description given throughout clause 15. Table XXXIV specifies the set of messages and states required to effect transition from one active state to another. Table XXXV specifies the messages that shall be sent and the device function interaction required while each state is active.

TABLE XXXIV – C mnemonics

Messages:		Interface states:	
pon	– power on	CIDS	– CONTROLLER IDLE STATE
rsc	– request system control	CADS	– CONTROLLER ADDRESSED STATE
rpp	– request parallel poll	CTRS	– CONTROLLER TRANSFER STATE
gts	– go to standby	CACS	– CONTROLLER ACTIVE STATE
tca	– take control asynchronously	CPWS	– CONTROLLER PARALLEL POLL WAIT STATE
tcs	– take control synchronously	CPPS	– CONTROLLER PARALLEL POLL STATE
sic	– send interface clear	CSBS	– CONTROLLER STANDBY STATE
sre	– send remote enable	CAWS	– CONTROLLER ACTIVE WAIT STATE
		CSWS	– CONTROLLER SYNCHRONOUS WAIT STATE
IFC	– INTERFACE CLEAR	CSRS	– CONTROLLER SERVICE REQUESTED STATE
ATN	– ATTENTION	CSNS	– CONTROLLER SERVICE NOT REQUESTED STATE
TCT	– TAKE CONTROL	SNAS	– SYSTEM CONTROL NOT ACTIVE STATE
		SACS	– SYSTEM CONTROL ACTIVE STATE
		SRIS	– SYSTEM CONTROL REMOTE ENABLE IDLE STATE
		SRNS	– SYSTEM CONTROL REMOTE ENABLE NOT ACTIVE STATE
		SRAS	– SYSTEM CONTROL REMOTE ENABLE ACTIVE STATE
		SIIS	– SYSTEM CONTROL INTERFACE CLEAR IDLE STATE
		SINS	– SYSTEM CONTROL INTERFACE CLEAR NOT ACTIVE STATE
		SIAS	– SYSTEM CONTROL INTERFACE CLEAR ACTIVE STATE
		CSHS	– CONTROLLER STANDBY HOLD STATE
		ACDS	– ACCEPT DATA STATE (AH FUNCTION)
		ANRS	– ACCEPTOR NOT READY STATE (AH FUNCTION)
		STRS	– SOURCE TRANSFER STATE (SH FUNCTION)
		TADS	– TALKER ADDRESSED STATE (T FUNCTION)
		SDYS	– SOURCE DELAY STATE (SH FUNCTION)

TABLE XXXV – C message outputs[1)]

C state	Remote message sent			Device function interaction
	ATN	IDY	Multiline	
CIDS	(F)	(F)	(NUL)	DF shall not send interface messages
CADS	(F)	(F)	(NUL)	DF shall not send interface messages
CACS	T	F	[2)]	DF can send interface messages
CPWS	T	T	(NUL)	DF shall not send interface messages
CPPS	T	T	(NUL)	DF can receive PPR message
CSBS	F	(F)	(NUL)	DF shall not send interface messages
CSWS	T	F or (F)	(NUL)	DF shall not send interface messages
CAWS	T	F	(NUL)	DF shall not send interface messages
CTRS	T	F	TCT	DF shall continue to sent TCT message
CSHS	F	(F)	(NUL)	DF shall not send interface messages
		IFC		
SIIS		(F)		None
SINS		F		
SIAS		T		
		REN		
SRIS		(F)		None
SRNS		F		
SRAS		T		
CSNS		None		No service requests exist
CSRS				DF notified of request for service

[1)] Message values sent are shown opposite only those states which affect them. Each major section of the table corresponds to a group of mutually exclusive states within the controller function.

[2)] Any multiline interface message listed in table XXXVIII. Although enabled by the Controller function these messages originate within the device functions.

15.3 *Controller state descriptions*

15.3.1 *CONTROLLER IDLE STATE (CIDS)*

15.3.1.1 In the CIDS state the C function relinquishes all of its interface control capabilities.

The C function powers on in the CIDS state.

15.3.1.2 In the CIDS state the ATN and IDY messages shall be sent passive false and the NUL message shall be sent passive true.

NOTE - The IDY message is coded on the ATN and EOI signal lines. Throughout 15.3, whenever the ATN message is sent true and the IDY message sent active or passive false, the EOI signal line is active or passive false.

15.3.1.3 The C function shall exit the CIDS state and enter the CONTROLLER ADDRESSED STATE (CADS) when either:

- the TAKE CONTROL (TCT) message (sent by the controller-in-charge) is true and the TADS and ACDS states are active;
- or the SYSTEM CONTROL INTERFACE CLEAR ACTIVE STATE (SIAS) is active.

NOTE - *The expression containing the TCT message is optional.*

15.3.2 *CONTROLLER ADDRESSED STATE (CADS)*

15.3.2.1 In the CADS state the C function is in the process of becoming the controller-in-charge of the interface but is waiting until the current controller stops sending the ATN message.

15.3.2.2 In the CADS state the ATN and IDY messages shall be sent passive false and the NUL messages shall be sent passive true.

15.3.2.3 The C function shall exit the CADS state and enter:

a) the CONTROLLER ACTIVE STATE (CACS) if the ATN message is false;

b) the CIDS state within t_4 if the IFC message is true and the SACS state is not active.

15.3.3 *CONTROLLER ACTIVE STATE (CACS)*

15.3.3.1 In the CACS state the C function enables the transfer of multiline interface messages from the device function(s) to the interface signal lines. These messages include device addresses, universal commands, or addressed commands. The SH function determines when the device function(s) may change the message content of the multiline messages being sent. However, message content is determined solely by the device function(s).

15.3.3.2 The ATN message shall be sent continuously true and the IDY message shall be sent continuously false while the CACS state is active under which conditions any of the multiline messages in table XXXVI may be sent by the device functions.

TABLE XXXVI

Universal commands (multiline)	Addresses
– LLO – DCL – SPE – SPD – PPU	– (LAD)[1] – (TAD)[2] – UNL
Addressed commands	Secondary commands
– GET – GTL – PPC – SDC – TCT	– (SAD)[3] – PPD – PPE

[1] Represents a listen address of a specific device (received as MLA).

[2] Represents a talk address of a specific device (received as MTA or OTA).

[3] Represents a secondary address of a specific device (received as MSA or OSA).

15.3.3.3 The C function shall exit the CACS state and enter:

a) the CONTROLLER TRANSFER STATE (CTRS) if the TCT message (sent by its own device-function, but received as a remote message) is true, the TADS state is (optionally) inactive, and the ACDS state is active;

b) the CONTROLLER PARALLEL POLL WAIT STATE (CPWS) if the *request parallel poll (rpp)* message is true and neither the STRS nor SDYS state is active;

c) the CONTROLLER IDLE STATE (CIDS) within t_4 if the IFC message is true and the SACS state is not active;

d) the CONTROLLER STANDBY STATE (CSBS) if the *go to standby (gts)* message is true, and neither the STRS nor the SDYS state is active.

15.3.4 *CONTROLLER PARALLEL POLL WAIT STATE (CPWS)*

15.3.4.1 In the CPWS state the C function is conducting a parallel poll over the interface but waiting for the DIO lines to settle.

15.3.4.2 In the CPWS state the IDY message shall be sent true and the NUL message must be sent passive true.

15.3.4.3 The C function shall exit the CPWS state and enter:

a) the CONTROLLER PARALLEL POLL STATE (CPPS) after a period of T_6 has elapsed;

b) the CIDS state within t_4 if the IFC message is true and the SACS state is not active;

c) the CAWS state if the *rpp* message is false.

15.3.5 *CONTROLLER PARALLEL POLL STATE (CPPS)*

15.3.5.1 In the CPPS state the C function is conducting a parallel poll and actively transferring PPR message values to the device function(s) as received via the interface signal lines.

15.3.5.2 In the CPPS state the IDY message shall be sent true and the NUL message shall be sent passive true.

15.3.5.3 The C function shall exit the CPPS state and enter:

a) the CAWS state if the *rpp* message is false;

b) the CIDS state within t_4 if the IFC message is true and the SACS state is not active.

15.3.6 *CONTROLLER STANDBY STATE (CSBS)*

15.3.6.1 In the CSBS state the C function is allowing two or more devices to transfer device-dependent messages over the interface.

15.3.6.2 In the CSBS state the ATN message shall be sent false, the IDY message shall be sent passive false, and the NUL message shall be sent passive true.

15.3.6.3 The C function shall exit the CSBS state and enter:

a) the CONTROLLER STANDBY HOLD STATE (CSHS) if the take control synchronously (*tcs*) message is true and the ANRS state is active;

b) the CONTROLLER SYNCHRONOUS WAIT STATE (CSWS) if the take control asynchronously (*tca*) message is true;

c) the CIDS state within t_4 if the IFC message is true and the SACS state is not active.

15.3.7 *CONTROLLER SYNCHRONOUS WAIT STATE (CSWS)*

15.3.7.1 In the CSWS state the C function is in the process of entering the CONTROLLER ACTIVE WAIT STATE (CAWS) but is waiting for a specified time (T_7) to make sure that the current active talker recognizes the ATN message being sent over the interface. If this state was entered via the *tcs* message, the device function(s) shall continue to send it true during this state.This causes the Acceptor Handshake Interface Function to continue sending the RFD message false over the interface, holding off transfer of the next data byte.

15.3.7.2 In the CSWS state the ATN message shall be sent true, the IDY message shall be sent active or passive false, and the NUL message shall be sent passive true.

15.3.7.3 The C function shall exit the CSWS state and enter:

a) the CAWS state after a period of T_7 has elapsed or if the TADS state is active;

b) the CIDS state within t_4 if the IFC message is true and the SACS state is not active.

15.3.8 *CONTROLLER ACTIVE WAIT STATE (CAWS)*

15.3.8.1 In the CAWS state the C function is waiting for a period of T_9 before entering the CACS state. This wait shall occur in order to guarantee that the EOI line has settled to its proper value and that no device is responding erroneously to what appears to be a parallel poll.

15.3.8.2 In the CAWS state the ATN message shall be sent true, the IDY message shall be sent false, and the NUL message shall be sent passive true.

15.3.8.3 The C function shall exit the CAWS state and enter:

a) the CACS state if the *rpp* message is false and a period of T_9 has elapsed;

b) the CPWS state if the *rpp* message is true;

c) the CIDS state within t_4 if the IFC message is true and the SACS state is not active.

15.3.9 *CONTROLLER TRANSFER STATE (CTRS)*

15.3.9.1 In the CTRS state the C function is sending the TCT addressed command to another device and is thus in the process of becoming idle.

15.3.9.2 In the CTRS state the ATN message shall be sent true, the IDY message shall be sent false, and the TCT message shall be sent true continuously.

15.3.9.3 The C function shall exit the CTRS and enter the CIDS state when either:

- the STRS state becomes inactive;
- or within t_4 if the IFC message is true and the SACS state is not active.

15.3.10 *CONTROLLER SERVICE REQUESTED STATE (CSRS)*

15.3.10.1 In the CSRS state the C function is notifying the device function(s) via a local message that at least one device on the interface is requesting service.

15.3.10.2 The CSRS state does not provide a remote message-sending capability.

15.3.10.3 The C function shall exit the CSRS state and enter the CONTROLLER SERVICE NOT REQUESTED STATE (CSNS) if the SRQ message is false.

15.3.11 *CONTROLLER SERVICE NOT REQUESTED STATE (CSNS)*

15.3.11.1 In the CSNS state the C function is notifying the device function(s) via a local message that no device on the interface is requesting service.

15.3.11.2 The CSNS state does not provide a remote message-sending capability.

15.3.11.3 The C function shall exit the CSNS state and enter the CSRS state if the SRQ message is true.

15.3.12 *SYSTEM CONTROL NOT ACTIVE STATE (SNAS)*

15.3.12.1 In the SNAS state the C function relinquishes all of its system control capabilities.

15.3.12.2 The SNAS state does not provide a remote message-sending capability.

15.3.12.3 The C function shall exit the SNAS state and enter the SACS state if the *request system control (rsc)* message is true.

15.3.13 *SYSTEM CONTROL ACTIVE STATE (SACS)*

15.3.13.1 In the SACS state the C function is allowed to exercise its system control capabilities.

15.3.13.2 The SACS state does not provide a remote message-sending capability.

15.3.13.3 The C function shall exit the SACS state and enter the SNAS state if the *rsc* message is false.

15.3.14 *SYSTEM CONTROL INTERFACE CLEAR IDLE STATE (SIIS)*

15.3.14.1 In the SIIS state the C function has no capability to clear the interface.

The controller interface function powers on in the SIIS state.

15.3.14.2 In the SIIS state the IFC message shall be sent passive false.

15.3.14.3 The C function shall exit the SIIS state if the SACS state is active and enter:

a) the SYSTEM CONTROL INTERFACE CLEAR NOT ACTIVE STATE (SINS) if the *send interface clear (sic)* message is false;

b) the SYSTEM CONTROL INTERFACE CLEAR ACTIVE STATE (SIAS) if the *sic* message is true.

15.3.15 *SYSTEM CONTROL INTERFACE CLEAR NOT ACTIVE STATE (SINS)*

15.3.15.1 In the SINS state the C function is not engaged in clearing the interface.

15.3.15.2 In the SINS state the IFC message shall be sent false continuously.

15.3.15.3 The C function shall exit the SINS state and enter:

a) the SIAS state if the local *sic* message is true;

b) the SIIS state if the SACS state is not active.

15.3.16 *SYSTEM CONTROL INTERFACE CLEAR ACTIVE STATE (SIAS)*

15.3.16.1 In the SIAS state the C function is engaged in clearing the interface.

All interface functions connected to the system shall respond to the IFC true message and will transfer to a known initial state.

15.3.16.2 In the SIAS state the IFC message shall be sent true.

15.3.16.3 The C function shall exit the SIAS state and enter:

a) the SINS state if the *sic* message is false and the SIAS state has been active for at least a period of T_8;

b) the SIIS state if the SACS state is not active.

15.3.17 *SYSTEM CONTROL REMOTE ENABLE IDLE STATE (SRIS)*

15.3.17.1 In the SRIS state the C function has no remote enable capability. All implementations of the C function should remain in the SRIS state continuously except when used in a device capable of System Controller performance.

The C function powers on in the SRIS state.

15.3.17.2 In the SRIS state the REN message shall be sent passive false.

15.3.17.3 The C function shall exit the SRIS state and enter:

a) the SYSTEM CONTROL REMOTE ENABLE NOT ACTIVE STATE (SRNS) if the *send remote enable (sre)* message is false and the SACS state is active;

b) the SYSTEM CONTROL REMOTE ENABLE ACTIVE STATE (SRAS) if the *sre* message is true, the SACS state is active and the SRIS state has been active for at least a period of T_8.

15.3.18 *SYSTEM CONTROL REMOTE ENABLE NOT ACTIVE STATE (SRNS)*

15.3.18.1 In the SRNS state the C function is not engaged in enabling remote operation of other devices over the interface.

15.3.18.2 In the SRNS state the REN message shall be sent passive false.

15.3.18.3 The C function shall exit the SRNS state and enter:

a) the SRAS state if the *sre* message is true for at least a period of T_8;

b) the SRIS state if the SACS state is not active.

15.3.19 *SYSTEM CONTROL REMOTE ENABLE ACTIVE STATE (SRAS)*

15.3.19.1 In the SRAS state the C function is actively engaged in enabling remote operation of other devices over the interface.

15.3.19.2 In the SRAS state the REN message shall be sent true continuously.

15.3.19.3 The C function shall exit the SRAS state and enter:

a) the SRNS state if the *sre* message is false;

b) the SRIS state if the SACS state is not active.

15.3.20 *CONTROLLER STANDBY HOLD STATE (CSHS)*

15.3.20.1 In the CSHS state the C function is in a hold state until such time as the DAV message is false as presented to all devices connected to the system. The CSHS state prevents false coincidence of the ATN and DAV messages, as observed by idle devices, during a *tcs sequence*.

15.3.20.2 In the CSHS state the ATN message shall be sent false, the IDY message shall be sent passive false and the NUL message shall be sent passive true.

15.3.20.3 The C function shall exit the CSHS state and enter:

a) the CONTROLLER STANDBY WAIT STATE (CSWS) if a period of T_{10} has elapsed;

b) the CONTROLLER STANDBY STATE (CSBS) if the *tcs* message is false;

c) the CONTROLLER IDLE STATE (CIDS) within t_4 if the IFC message is true and the SACS state is not active.

15.4 *Controller interface function allowable subsets*

The only allowable subsets to the controller interface function shall be those listed in table XXXVII.

15.5 *Additional C function requirements and guidelines*

15.5.1 Warning: Use *tca* with caution.

15.5.1.1 Restriction on the use *tca:*

The designer shall not assume that valid data will be transferred across the interface if the *tca* message becomes true while a device-dependent message is true.

15.5.1.2 Background:

Asynchronous interruption of an active talker by a controller through the use of *tca* may occur at any time when a device-dependent message is true. If a device-dependent message is true and ATN becomes true, the interrupted byte could be either lost or misinterpreted by other devices as an interface message (e.g. command or address) and produce unintended state transitions.

15.5.2 The *tcs* message, if used, may change from false to true only during the CSBS state. It may change from true to false only during the CAWS state. These restrictions guarantee that RFD is held false for the proper amount of time during a synchronous take-control operation. The CSBS to CSWS transition, via *tcs* and ANRS term, assumes that the device with the controller function participates actively as a listener in LACS.

TABLE XXXVII – Controller function allowable subsets

Identification[3]	Capabilities										Notes	States required									Other requirements			Other function subsets required				
	System controller	Send IFC and Take Charge	Send REN	Respond to SRQ	Send I.F. Messages	Receive Control	Pass Control	Pass Control to Self	Parallel Poll	Take Control Synchronously		SNAS, SACS	SIIS, SIAS, SINS	SRIS, SRAS, SRNS	CSNS, CSRS	CACS, CSBS, CSHS, CSWS, CAWS	CADS	CIDS	CTRS	CPWS, CPPS	$[\text{TCT} \wedge \boxed{\text{ACDS}} \wedge \boxed{\text{TADS}}]$[1]	$[\boxed{\text{TADS}}]$[2]	*tcs* not always false	C1	C2	SH1	AH1, L1-L4 or LE1-LE4	T1-T8, TE1-TE8
C0	N	N	N	N	N	N	N	N	N	N		O	O	O	O	O	O	O	O	O	O	O	O	O	–	–	–	–
C1	Y	–	–	–	–	–	–	–	–	–	1	R	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–
C2	–	Y	–	–	–	–	–	–	–	–	1	–	R	–	–	–	–	–	–	–	–	–	–	R	–	–	–	–
C3	–	–	Y	–	–	–	–	–	–	–	1	–	–	R	–	–	–	–	–	–	–	–	–	R	–	–	–	–
C4	–	–	–	Y	–	–	–	–	–	–	1	–	–	–	R	–	–	–	–	–	–	–	–	–	–	–	–	–
C5	–	–	–	–	Y	Y	Y	Y	Y	Y	2, 3	–	–	–	–	R	R	R	R	R	R	R	R	–	–	R	R	R
C6	–		–	–	Y	Y	Y	Y	Y	N	2, 3	–	–	–	–	R	R	R	R	R	R	R	O	–	–	R	–	R
C7	–	–	–	–	Y	Y	Y	Y	N	Y	2, 3	–	–	–	–	R	R	R	R	O	R	R	R	–	–	R	R	R
C8	–	–	–	–	Y	Y	Y	Y	N	N	2, 3	–	–	–	–	R	R	R	R	O	R	R	O	–	–	R	–	R
C9	–	–	–	–	Y	Y	Y	N	Y	Y	2, 3	–	–	–	–	R	R	R	R	R	R	O	R	–	–	R	R	R
C10	–	–	–	–	Y	Y	Y	N	Y	N	2, 3	–	–	–	–	R	R	R	R	R	R	O	O	–	–	R	–	R
C11	–	–	–	–	Y	Y	Y	N	N	Y	2, 3	–	–	–	–	R	R	R	R	O	R	O	R	–	–	R	R	R
C12	–	–	–	–	Y	Y	Y	N	N	N	2, 3	–	–	–	–	R	R	R	R	O	R	O	O	–	–	R	–	R
C13	–	–	–	–	Y	Y	N	N	Y	Y	2	–	–	–	–	R	R	R	O	R	O	O	R	–	R	R	R	–
C14	–	–	–	–	Y	Y	N	N	Y	N	2	–	–	–	–	R	R	R	O	R	O	O	O	–	R	R	–	–
C15	–	–	–	–	Y	Y	N	N	N	Y	2	–	–	–	–	R	R	R	O	O	O	O	R	–	R	R	R	–
C16	–	–	–	–	Y	Y	N	N	N	N	2	–	–	–	–	R	R	R	O	O	O	O	O	–	R	R	–	–
C17	–	–	–	–	Y	N	Y	Y	Y	Y	2, 3, 4	–	–	–	–	R	O	R	R	R	R	R	R	–	–	R	R	R
C18	–	–	–	–	Y	N	Y	Y	Y	N	2, 3, 4	–	–	–	–	R	O	R	R	R	R	R	O	–	–	R	–	R
C19	–	–	–	–	Y	N	Y	Y	N	Y	2, 3, 4	–	–	–	–	R	O	R	R	O	R	R	R	–	–	R	R	R
C20	–	–	–	–	Y	N	Y	Y	N	N	2, 3, 4	–	–	–	–	R	O	R	R	O	R	R	O	–	–	R	–	R
C21	–	–	–	–	Y	N	Y	N	Y	Y	2, 3, 4	–	–	–	–	R	O	R	R	R	R	O	R	–	–	R	R	R
C22	–	–	–	–	Y	N	Y	N	Y	N	2, 3, 4	–	–	–	–	R	O	R	R	R	R	O	O	–	–	R	–	R
C23	–	–	–	–	Y	N	Y	N	N	Y	2, 3, 4	–	–	–	–	R	O	R	R	O	R	O	R	–	–	R	R	R
C24	–	–	–	–	Y	N	Y	N	N	N	2, 3, 4	–	–	–	–	R	O	R	R	O	R	O	O	–	–	R	–	R
C25	–	–	–	–	Y	N	N	N	Y	Y	2	–	–	–	–	R	O	O	O	R	O	O	R	–	–	R	R	–
C26	–	–	–	–	Y	N	N	N	Y	N	2	–	–	–	–	R	O	O	O	R	O	O	O	–	–	R	–	–
C27	–	–	–	–	Y	N	N	N	N	Y	2	–	–	–	–	R	O	O	O	O	O	O	R	–	–	R	R	–
C28	–	–	–	–	Y	N	N	N	N	N	2	–	–	–	–	R	O	O	O	O	O	O	O	–	–	R	–	–

O omit
R required
– not applicable or not required
Y yes
N no

1) This is part of the CIDS to CADS transitional expression.
2) This is part of the CACS to CTRS transitional expression.
3) Typical notation to describe a controller consists of the letter C followed by one or more of the numbers indicating the subsets selected. For example: C1, 2, 3, 4, 8.

NOTES

1 One or more of subsets C1 to C4 may be chosen in any combination with any *one* of subsets C5 to C28.

2 Only one subset may be chosen from C5 to C28.

3 The CTRS state shall be included in devices which are to be operated in multi-controller systems.

4 These subsets are not allowed unless C2 is included.

16 Remote message coding and transfer

16.1 *Remote message coding*

Each remote message is sent by an interface function and received by an interface function via one or more interface signal lines. This clause defines the complete set of remote messages and how they are coded and transferred on the signal lines. The coding of all remote messages sent or received by the various interface functions is specified in the remote message coding table XXXVIII.

16.2 *Remote message coding concepts*

16.2.1 Messages may be coded into the logical state of one or more signal lines.

16.2.2 For this part a message derived from or sent as the logical state of only one signal line is referred to as a uniline message (e.g. ATN).

16.2.3 For this part a message derived from or sent as a combination of logical states of two or more signal lines is referred to as a multiline message (e.g. DCL).

16.2.4 A message may be defined as a logical combination (AND, OR, or NOT) of other messages (e.g. OTA).

16.2.5 The *coding* of a message sent and received is the same.

16.3 *Remote message transfer*

16.3.1 A message is sent by driving one or more specified signal lines to a logical 1 or a logical Ø. Lines not specified as part of the message coding shall not be driven.

16.3.2 A message is received by sensing one or more specified bus signal lines to determine the logical value of each signal line as either 1 or Ø. Lines not specified as part of the message coding are ignored.

16.3.3 A uniline message value is considered valid as soon as its corresponding logical value is detected. (See tables III, VI, IX, XIX, XXV and XXXV for times at which messages are sent.)

16.3.4 A multiline message is valid only within the context of the Source and Acceptor Handshake Functions. A transmitted multiline message is valid while the Source Handshake Function is in the Source Transfer (STRS) state. A received multiline message is valid while the Acceptor Handshake Function is in the Accept Data (ACDS) state.

16.3.5 All passive message values are transferred as ∅ signal line states. This requires only the logic OR of signal line states to be performed on the interface.

16.4 *Remote message coding table organization and conventions*

16.4.1 All messages capable or being sent or received by an interface function are listed by name and mnemonic.

16.4.2 The table correlates the message value (true or false) to the bus signal line logical value (1 or ∅) and vice versa.

16.4.3 Each remote message entry in the table specifies both the encoding required to send the messages and the decoding required to receive the messages.

16.4.4 The true value of a uniline message is specified by the assignment of a specific logical value to a signal line.

NOTE - Some uniline messages (e.g. SRQ) may change at many different times and other uniline messages may change only at specific times (e.g. END).

16.4.5 The true value of a multiline message is specified by the assignment of a unique set of logical values (1 or ∅) to the corresponding set of signal lines that contain the message.

16.4.6 The false value of a message is any combination of logical values (1 or ∅) other than the unique set that specifies the true value.

16.4.7 Each message entry in the table is identified by type: either uniline (U) or multiline (M). Each message is further identified by class (1 of 7) according to the function it performs within the interface function or device function.

16.4.8 The logical value that a bus signal line may have is specified in the table as a ∅, 1, X or Y. These represent the logical values as follows:

- ∅ logical zero
- 1 logical one
- X don't care (for the coding of a *received* message)
- X shall not drive unless directed by another message (for the coding of a *transmitted* message)
- Y don't care (for the coding of a *transmitted* message)

16.5 *Remote message coding table perspective*

Table XXXVIII is constructed to reflect each remote message sent (or received) by each interface function. In actual use, two or more messages as defined in the table may be sent concurrently (e.g. DAB true and ATN false) by *different* interface functions (see notes 9 and 10 to table XXXVIII and annex D).

16.6 *ISO code representation: message coding guidelines*

Many devices use the ISO 7-bit code (ISO 646) because it is convenient both to generate and interpret this code. The relationships between the ISO code and the messages (binary bit patterns) defined and described in this part are identified in this clause.

16.6.1 *Interface messages*

The interface system utilizes message coding as defined in table XXXVIII to carry interface messages among devices when the ATN message is true. This coding may be correlated to the ISO code by relating DIO1 through DIO7 to bits 1 to 7, respectively. The ISO code does not contain the equivalent of the dedicated ATN message (bit or line).

When the interface system defined in this part is interconnected, via a terminal unit, to other environments, then protocol beyond the scope of this part shall be used to enable proper communication and avoid possible contradictions with other assigned meanings for the ISO code.

TABLE XXXVIII – Remote message coding

Mnemonic	Message name		Type	Class	Bus signal line(s) and coding that asserts the true value of the message									
					DIO 8 7	6 5 4	DIO 3 2 1	DAV NRFD NDAC	ATN	EOI	SRQ	IFC	REN
ACG	ADDRESSED COMMAND GROUP		M	AC	Y0	00X	XXX	XXX	1	X	X	X	X
ATN	ATTENTION		U	UC	XX	XXX	XXX	XXX	1	X	X	X	X
DAB	DATA BYTE	(Notes 1, 9)	M	DD	DD	DDD	DDD	XXX	0	X	X	X	X
					8 7	6 5 4	3 2 1						
DAC	DATA ACCEPTED		U	HS	XX	XXX	XXX	XX0	X	X	X	X	X
DAV	DATA VALID		U	HS	XX	XXX	XXX	1XX	X	X	X	X	X
DCL	DEVICE CLEAR		M	UC	Y0	010	100	XXX	1	X	X	X	X
END	END	(Note 9)	U	ST	XX	XXX	XXX	XXX	0	1	X	X	X
EOS	END OF STRING	(Notes 2, 9)	M	DD	EE	EEE	EEE	XXX	0	X	X	X	X
					8 7	6 5 4	3 2 1						
GET	GROUP EXECUTE TRIGGER		M	AC	Y0	001	000	XXX	1	X	X	X	X
GTL	GO TO LOCAL		M	AC	Y0	000	001	XXX	1	X	X	X	X
IDY	IDENTIFY		M	UC	XX	XXX	XXX	XXX	1	1	X	X	X
IFC	INTERFACE CLEAR		U	UC	XX	XXX	XXX	XXX	X	X	X	1	X
LAG	LISTEN ADDRESS GROUP		M	AD	Y0	1XX	XXX	XXX	1	X	X	X	X
LLO	LOCAL LOCKOUT		M	UC	Y0	010	001	XXX	1	X	X	X	X
MLA	MY LISTEN ADDRESS	(Note 3)	M	AD	Y0	1LL	LLL	XXX	1	X	X	X	X
						5 4	3 2 1						
MTA	MY TALK ADDRESS	(Note 4)	M	AD	Y1	0TT	TTT	XXX	1	X	X	X	X
						5 4	3 2 1						
MSA	MY SECONDARY ADDRESS	(Note 5)	M	SE	Y1	1SS	SSS	XXX	1	X	X	X	X
						5 4	3 2 1						
NUL	NULL BYTE		M	DD	00	000	000	XXX	X	X	X	X	X
OSA	OTHER SECONDARY ADDRESS		M	SE	$(OSA = SCG \wedge \overline{MSA})$								
OTA	OTHER TALK ADDRESS		M	AD	$(OTA = TAG \wedge \overline{MTA})$								
PCG	PRIMARY COMMAND GROUP		M	–	$(PCG = ACG \vee UCG \vee LAG \vee TAG)$								
PPC	PARALLEL POLL CONFIGURE		M	AC	Y0	000	101	XXX	1	X	X	X	X
PPE	PARALLEL POLL ENABLE	(Note 6)	M	SE	Y1	10S	PPP	XXX	1	X	X	X	X
							3 2 1						
PPD	PARALLEL POLL DISABLE	(Note 7)	M	SE	Y1	11D	DDD	XXX	1	X	X	X	X
						4	3 2 1						
PPR1	PARALLEL POLL RESPONSE 1	(Note 10)	U	ST	XX	XXX	XX1	XXX	1	1	X	X	X
PPR2	PARALLEL POLL RESPONSE 2		U	ST	XX	XXX	X1X	XXX	1	1	X	X	X
PPR3	PARALLEL POLL RESPONSE 3		U	ST	XX	XXX	1XX	XXX	1	1	X	X	X
PPR4	PARALLEL POLL RESPONSE 4		U	ST	XX	XX1	XXX	XXX	1	1	X	X	X
PPR5	PARALLEL POLL RESPONSE 5		U	ST	XX	X1X	XXX	XXX	1	1	X	X	X
PPR6	PARALLEL POLL RESPONSE 6		U	ST	XX	1XX	XXX	XXX	1	1	X	X	X
PPR7	PARALLEL POLL RESPONSE 7		U	ST	X1	XXX	XXX	XXX	1	1	X	X	X
PPR8	PARALLEL POLL RESPONSE 8		U	ST	1X	XXX	XXX	XXX	1	1	X	X	X
PPU	PARALLEL POLL UNCONFIGURE		M	UC	Y0	010	101	XXX	1	X	X	X	X
REN	REMOTE ENABLE		U	UC	XX	XXX	XXX	XXX	X	X	X	X	1
RFD	READY FOR DATA		U	HS	XX	XXX	XXX	X0X	X	X	X	X	X
RQS	REQUEST SERVICE	(Notes 8, 9)	U	ST	X1	XXX	XXX	XXX	0	X	X	X	X
SCG	SECONDARY COMMAND GROUP		M	SE	Y1	1XX	XXX	XXX	1	X	X	X	X
SDC	SELECTED DEVICE CLEAR		M	AC	Y0	000	100	XXX	1	X	X	X	X
SPD	SERIAL POLL DISABLE		M	UC	Y0	011	001	XXX	1	X	X	X	X
SPE	SERIAL POLL ENABLE		M	UC	Y0	011	000	XXX	1	X	X	X	X
SRQ	SERVICE REQUEST		U	ST	XX	XXX	XXX	XXX	X	X	1	X	X
STB	STATUS BYTE	(Notes 8, 9)	M	ST	SX	SSS	SSS	XXX	0	Y	X	X	X
					8	6 5 4	3 2 1						
TCT	TAKE CONTROL		M	AC	Y0	001	001	XXX	1	X	X	X	X
TAG	TALK ADDRESS GROUP		M	AD	Y1	0XX	XXX	XXX	1	X	X	X	X
UCG	UNIVERSAL COMMAND GROUP		M	UC	Y0	01X	XXX	XXX	1	X	X	X	X
UNL	UNLISTEN		M	AD	Y0	111	111	XXX	1	X	X	X	X
UNT	UNTALK	(Note 11)	M	AD	Y1	011	111	XXX	1	X	X	X	X

(continued on page 73)

TABLE XXXVIII *(concluded)*

Notes and symbols for remote message coding table XXXVIII

Level assignment:

- **ø high-state signal level**
- **1 low-state signal level**

The coding of table XXXVIII may be translated to equivalent electrical signal levels as specified in clause 18.

Symbols:

Type:	**U**	**uniline message**
	M	**multiline message**
Class:	**AC**	**addressed command**
	AD	**address (talk or listen)**
	DD	**device-dependent**
	HS	**handshake**
	UC	**universal command**
	SE	**secondary**
	ST	**status**

NOTES

1 D1-D8 specify the device-dependent data bits.

2 E1-E8 specify the device-dependent code used to indicate the EOS message. This message may be used to terminate a string of DAB messages.

3 L1-L5 specify the device-dependent bits of the device's listen address (see 39.2.1).

4 T1-T5 specifiy the device-dependent bits of the device's talk address (see 39.1.1).

5 S1-S5 specify the device-dependent bits of the device's secondary address (see 39.3.1).

6 S specifies the sense of the parallel poll response.

S	*Response*
ø	ø
1	1

P1-P3 specify the parallel poll response (PPR) message to be sent when a parallel poll is executed.

P3	*P2*	*P1*	*PPR message*
ø	ø	ø	PPR1
•	•	•	•
•	•	•	•
•	•	•	•
1	1	1	PPR8

7 D1-D4 specify don't-care bits that shall not be decoded by the receiving device. It is recommended that all zeros be sent.

8 S1-S6, S8 specify the device-dependent status.

(NOTE - DIO7 is used for the RQS message).

9 The source of the message on the ATN line is always the C function, whereas the messages on the DIO and EOI lines are enabled by the T function.

10 The source of the messages on the ATN and EOI lines is always the C function, whereas the source of the messages on the DIO lines is always the PP function.

16.6.2 *Device-dependent messages*

The specific coding of device-dependent messages is beyond the scope of this part. After a talker and listener(s) have been addressed via interface messages, any commonly understood binary code of 8 bits or less, e.g. BCD, or alphanumeric code may be used when the ATN message is false.

- The alphanumeric codes (dense subset of the ISO code, columns 2 to 5) are preferred for communication of the device-dependent messages wherever possible. Bit 1 to bit 7 of the ISO code correspond to DIO1-DIO7.
- When other codes are used (e.g. binary) the most significant bit should be placed on the DIO line that has the highest number (e.g. DIO8 for bit 8).

16.6.3 The ISO code is further illustrated in annex E as it correlates with the codes of this part.

16.7 *State transition timing values*

The T_x and t_y values listed in section 2 throughout the interface function descriptions and state diagrams are defined in section 3, clause 24.

SECTION 3: ELECTRICAL SPECIFICATIONS

17 Application

This section defines the electrical specifications for interface systems to be used in environments where:

- physical distance between devices is short;
- electrical noise is relatively low.

All electrical specifications for the driver and receiver circuits are based on the use of TTL technology.

NOTES

1 The interface function circuits connected to the drivers or receivers may be implemented in other device technologies at the designer's choice.

2 Driver and receiver devices need only be used on those signal lines required for the interface functions implemented (see clause 21 for termination requirements).

3 Either open collector or three-state drivers may be used as determined by data rate considerations of clauses 19 and 31.

18 Logical and electrical state relationships

18.1 The relationship between the logical states defined in the Remote Message Coding table XXXVIII and the electrical state levels present on the signal lines is as follows:

Coding logical state	*Electrical signal levels*
Ø	corresponds to ≥+2,0 V called "high state"
1	corresponds to ≤+0,8 V called "low state"

The high and low states are based on standard TTL levels for which the power source does not exceed +5,25 V d.c. and is referred to logic ground.

18.2 This section indicates current flow into a node with a positive sign and current flow out of a node with a negative sign.

19 Driver requirements

Messages may be sent in either an active or passive manner over the interface (see 4.3). All passive true message transfer occurs in the high state and shall be carried on a signal line using open collector drivers.

19.1 *Driver types*

19.1.1 Open collector drivers shall be used on the SRQ, NRFD, NDAC signal lines.

19.1.2 Open collector drivers or three-state drivers may be used on the DIO 1-8, DAV, IFC, ATN, REN, and EOI signal lines with this exception: DIO 1-8 shall use open collector drivers for parallel polling applications (see 12.3.3).

NOTE - Three-state drivers are useful for systems where higher speed operations is required.

19.1.3 It is recommended that a three-state driver be used within a controller to drive the ATN signal line if the controller is intended to be used in a system in which other devices are implemented with three-state drivers on the DIO, DAV, and EOI signal lines.

19.2 *Driver specifications*

The specification for drivers shall be as follows:

Low state: output voltage (three-state or open-collector drivers) ≤ +0,5 V at +48 mA sink current.

The driver shall be capable of sinking 48 mA continuously.

High state: output voltage (three-state) ≥ +2,4 V at –5,2 mA;
output voltage (open-collector) (see 21.5).

The indicated voltage values are measured at the device connector between the signal line and logic ground.

See clause 21 for additional requirements which may apply to the driver.

20 Receiver requirements

20.1 *Receiver specifications, allowed*

The specification for receivers with nominal noise immunity shall be as follows:

Input voltage ≤ +0,8 V for the low state;

Input voltage ≥ +2,0 V for the high state.

See 21.3 for additional requirements which may apply to the receiver.

20.2 *Receiver specifications, preferred*

To provide greater noise immunity, the use of Schmitt-type receiver circuits or equivalent for all signal lines is recommended.

The specifications for these receivers shall be as follows:

Low state:	negative going threshold voltage ≥ +0,8 V;
High state:	positive going threshold voltage ≤ +2,0 V;
Hysteresis:	≥ 0,4 V.

21 Composite device load requirements

21.1 *Resistive termination*

Each signal line (whether or not it is connected to a driver or receiver) shall be terminated within the device by a resistive load whose major purpose is to establish a steady-state voltage when all drivers on a line are in the high-impedance state. This load is also used to maintain a uniform device impedance on the line and improve noise immunity. For specific requirements, see 21.3.3, and for typical resistive values, see 21.5.

21.2 *Negative voltage clamping*

Each signal line to which a receiver is connected shall be provided with means to limit the negative voltage excursions. Typically this circuit element is a diode clamp contained within the receiver component.

21.3 *DC load requirements*

21.3.1 The d.c. load characteristics of a device are affected by the driver and receiver circuits as well as the resistive termination and voltage clamping circuits; therefore they are specified for the composite device interface circuit, *not* the individual components. This clause, however, provides complete specifications for the resistive termination and voltage clamping circuits.

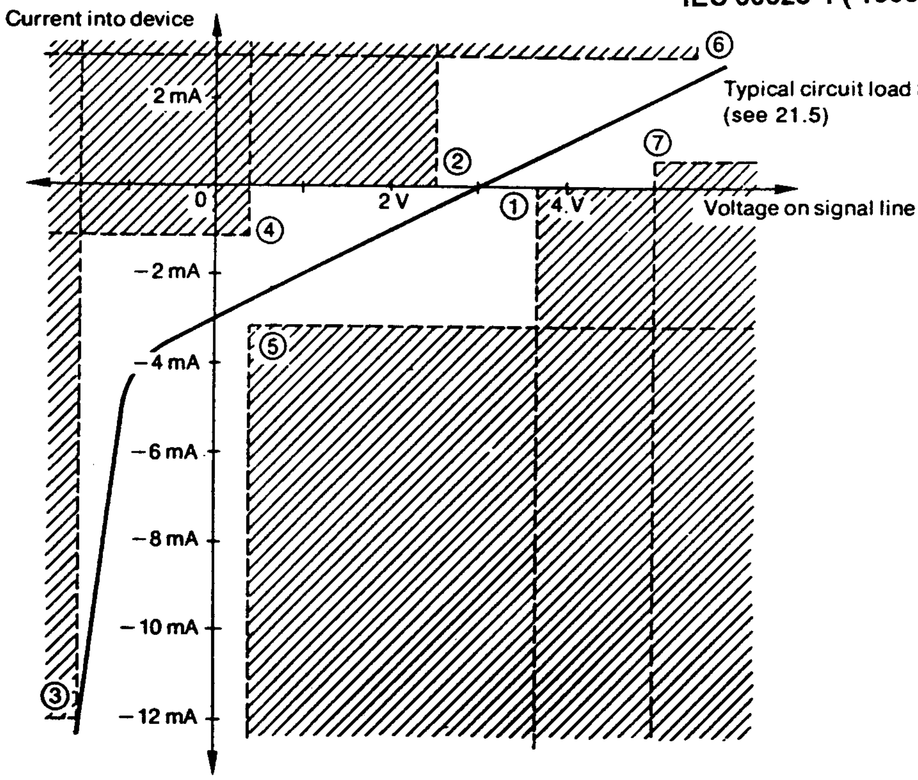
21.3.2 Load measurement conditions assume that the receiver, driver, and resistive termination circuits are connected together within the device with the driver in the high-impedance state.

21.3.3 Each signal line interface within a device shall have the following d.c. load characteristics and shall fall within the unshaded area of figure 15.

1:	if I ≤	0 mA,	U shall be <3,7 V
2:	if I ≥	0 mA,	U shall be >2,5 V
3:	if I ≥	−12,0 mA,	U shall be >−1,5 V (only if receiver exists)
4:	if U ≤	0,4 V,	I shall be <−1,3 mA
5:	if U ≥	0,4 V,	I shall be >3,2 mA
6:	if U ≤	5,5 V,	I shall be <2,5 mA
7:	if U ≥	5,0 V,	I shall be >0,7 mA or the small-signal impedance Z shall be ≤2kΩ at 1 MHz

21.4 *Capacitive load limit*

The internal capacitance load on each signal line shall not exceed 100 pF within each device, and should be substantially to ground.

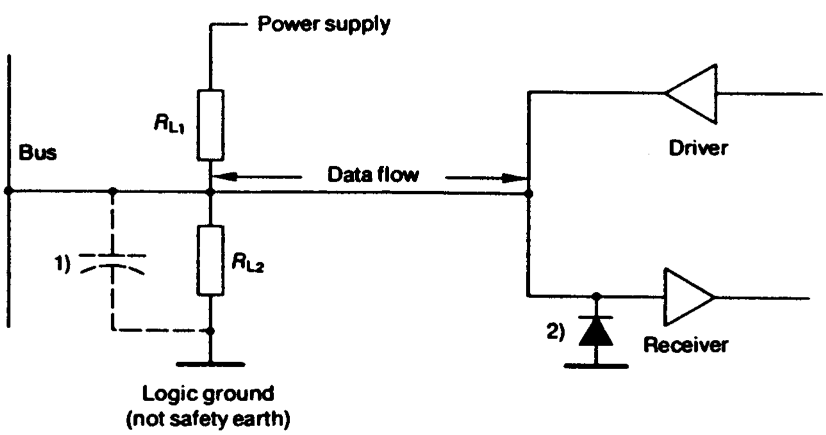


NOTE - The slope of the d.c. load line should, in general, correspond to a resistance not in excess of 3 kΩ.

Figure 15 – DC load boundary specification

21.5 *Typical circuit configuration*

Figure 16 shows a typical circuit configuration for signal line input/output circuits for which readily available components exist.



1) Stray capacitance allowed by 21.4.

2) Typically contained within receiver component.

Figure 16 – Typical signal line input/output circuits

This basic circuit is compatible with both TTL microcircuit and discrete element devices.

The specifications for this typical configuration are as follows:

R_{L1}: 3 kΩ ± 5 %
R_{L2}: 6,2 kΩ ± 5 %

Driver: output leakage current = +0,25 mA max. at U_z = +5,25 V (if open-collector driver is used);

output leakage current = ±40 μA max. at U_z = +2,4 V (if three-state driver is used).

Receiver: input current = −1,6 mA max. at U_z = +0,4 V;
input leakage current = +40 μA max. at U_z = +2,4 V;
= +1,0 mA max. at U_z = +5,25 V.

Power supply voltage = +5 V ± 5 %.

Only a single driver and receiver may be connected to each signal line in the typical configuration of figure 16. Other configurations may exist in which this restriction does not hold provided the composite device load specifications of 21.3 are met.

22 Ground requirements

22.1 The overall shield of the interconnecting cable shall be connected through one contact of the connector to frame (safety earth) to minimize susceptibility to and generation of external noise.

Warning: Devices should not be operated at significantly different frame potentials. The interface connection system may not be capable of handling excessive ground currents.

22.2 It is recommended that the ground returns of the individual control and status signal lines be connected to logic ground at the logic circuit driver or receiver to minimize crosstalk interference transients.

22.3 A removable link to connect the logic ground to frame is recommended if the frame is connected to the metal enclosure or the protective earth terminal, in order to avoid ground loops.

23 Cable characteristics

23.1 *Conductor requirements*

The maximum resistance for the cable conductors shall be, per metre of length:

- each signal line (e.g. DIO1, ATN) 0,14 Ω
- each individual signal line ground return 0,14 Ω
- common logic ground return 0,085 Ω
- overall shield 0,0085 Ω

23.2 *Cable construction*

23.2.1 The cable should contain an overall shield and at least 24 conductors of which 16 shall be used for signal lines and the balance used for logic ground returns.

23.2.2 The maximum capacitance measured (at 1 kHz) between any signal line and *all* other lines (signals, grounds, and shield) connected to ground shall be 150 pF per meter.

23.2.3 The shield shall contain a braid with at least 85 % coverage.

23.2.4 The cable shall be constructed to minimize the effects of cross-talk between signal lines, the susceptibility of the signal lines to external noise, and the transmission of interface signals to the external environment.

- Each of the signal lines DAV, NRFD, NDAC, EOI, ATN, IFC, REN and SRQ shall be twisted with one of the logic ground wires or isolated using an equivalent scheme (to minimize crosstalk).
- A cable construction in which twisted pairs are contained in the core of the cable and the individual DIO lines contained around the periphery of this core has been found satisfactory as has been the use of twisted pair conductors for all 16 signal lines where each original line is twisted with an earth conductor.
- The cable shall contain an overall shield carried through the cable assembly and connectors at both ends.
- Alternately, any other internal cable construction which yields the same results may be used.

24 State transition time value

To ensure maximum possible compatibilty among interconnected devices, table XXXIX states the mandatory time relationships between critical signal inputs and outputs to a specific device. The time values T_1, T_6-T_9 allow for the normal propagation delays of the transmission path and the typical circuit delays within other devices. They are measured from the time the source output driver is seen to start its transition as viewed from its associated connector.

Care should be taken when the values T_1, T_6-T_9 approach the minimum values specified in table XXXIX. In this case voltage values, capacitance and crosstalk become more critical. It is recommended that:

- the high-state driver voltage be maintained at its full value;
- cable resistance and capacitance be kept as low as possible;
- crosstalk be kept at a minimum value.

TABLE XXXIX

Time values			
Time value identifier[1]	Function (applies to)	Description	Value
T_1	SH	Setting time for multiline messages	≥ 2 µs[2]
t_2	SH, AH, T, L, TE, LE	Response to ATN	≤ 200 ns
T_3	AH	Interface message accept time[3]	> 0[4]
t_4	T, TE, L, LE, C, RL	Response to IFC or REN false	< 100 µs
t_5	PP	Response to ATN ∧ EOI	≤ 200 ns
T_6	C	Parallel poll execution time	≥ 2 µs
T_7	C	Controller delay to allow current talker to see ATN message	≥ 500 ns
T_8	C	Length of IFC or REN false	> 100 µs
T_9	C	Delay for EOI[5]	≥ 1,5 µs[6]
T_{10}	C	Delay for $\overline{\text{DAV}}$	≥ 1,5 µs

1) Time values specified by a lower-case t indicate the maximum time allowed to make a state transition. Time values specified by an upper-case T indicate the minimum time that a function shall remain in a state before exiting.

2) If three-state drivers are used on the DIO, DAV, and EOI lines, T_1 may be:

a) ≥ 1 100 ns,

b) ≥ 700 ns if it is known that within the controller ATN is driven by a three-state driver, however this value is not recommended,

c) or ≥ 500 ns for all subsequent bytes following the first sent after each false transition of ATN (the first byte shall be sent in accordance with a) or b) above),

d) or ≥ 350 ns for all subsequent bytes following the first sent after each false transition of ATN under conditions specified in 31.3.

3) Time required for interface functions to accept, not necessarily respond to, interface messages.

4) Implementation-dependent.

5) Delay required for EOI, NDAC, and NRFD signal lines to indicate valid states.

6) ≥ 600 ns for three-state drivers.

SECTION 4: MECHANICAL SPECIFICATIONS

25 Application

This section defines the specification for interface systems to be used in environments where:

- physical distances between devices are limited;
- star or linear bus interconnection networks are useful;
- connector mounting space is limited.

26 Connector type

A quality connector of the rack and panel type with proven performance shall be used which has these minimum characteristics.

26.1 *Electrical requirements*

Voltage rating:		200 V
Current rating:	(at T = 25 °C)	5 A per contact
Contact resistance:	(at 10 mA)	less than 20 mΩ
Insulation resistance		greater than 1 GΩ
Test voltage:	(1 min, 20 °C)	500 V
Capacitance:	(between contacts at 1 kHz)	< 1,5 pF
Endurance:	(with 1 A and 70 °C)	> 1 000 h

26.2 *Mechanical requirements* (see figure 17)

Number of contacts:	24*
Contact pitch: (self-wiping)	2,16 mm
Polarization: (shell shape)	trapezoidal
Shell material:	corrosion resistant, conductive
Retention force per contact: (as for IEC 603-2)	> 0,15 N
Typical insertion and withdrawal force (F):	8 N < F < 89 N
Endurance: (for specified contact resistance)	> 500 insertions
Clearance between adjacent contacts:	> 0,5 mm
Solderability (if applicable):	as for IEC 512 test, 12a, 235 °C, 2 sec

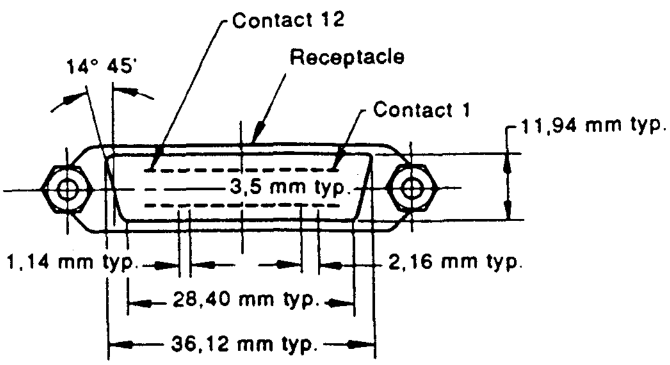


Figure 17 – Connector: receptacle dimensions

26.3 *Environmental requirements*

Basic environmental performance relative to the cold, dry heat and damp-heat tests for climatic category 25/070/21 as explained in annex A of IEC 68-1.

* See annex K for description of alternate 25-pin connector type.

27 Connector contact assignment

The contact assigment of the cable connector and the device connector shall be as shown below:

Contact	Signal line	Contact	Signal line
1	DIO 1	13	DIO 5
2	DIO 2	14	DIO 6
3	DIO 3	15	DIO 7
4	DIO 4	16	DIO 8
5	EOI (24)	17	REN (24)
6	DAV	18	Gnd (6)
7	NRFD	19	Gnd (7)
8	NDAC	20	Gnd (8)
9	IFC	21	Gnd (9)
10	SRQ	22	Gnd (10)
11	ATN	23	Gnd (11)
12	SHIELD	24	Gnd LOGIC

NOTE - "Gnd (*n*)" refers to the signal ground return of the referenced contact. EOI and REN return on contact 24.

28 Device connector mounting

28.1 Each device shall be provided with a receptacle type connector having the typical dimensions shown in figure 17 and with which the plug type connector shall mate. The two rows, each of 12 contacts, are centered within the trapezoidal shell. The connector mounting shall make provisions to accept the locking screws of the cable assembly.

The preferred orientation of the connector, as mounted on a device and viewed from the rear of the device in its normal operating position, is with contact 1 in the upper right-hand corner. The connector location should allow for sufficient cable clearance as shown in figure 21.

28.2 The connector may be mounted on either the outside or inside of the panel for which the typical panel cut-out dimensions are given in figure 18.

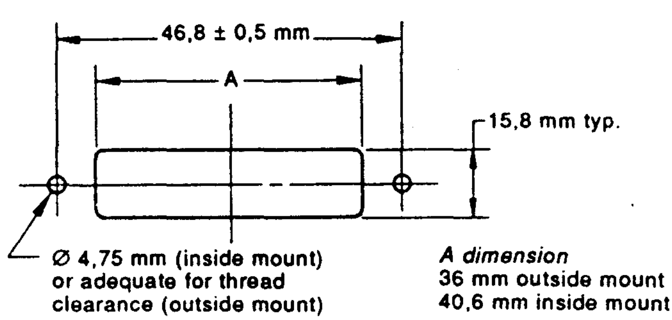
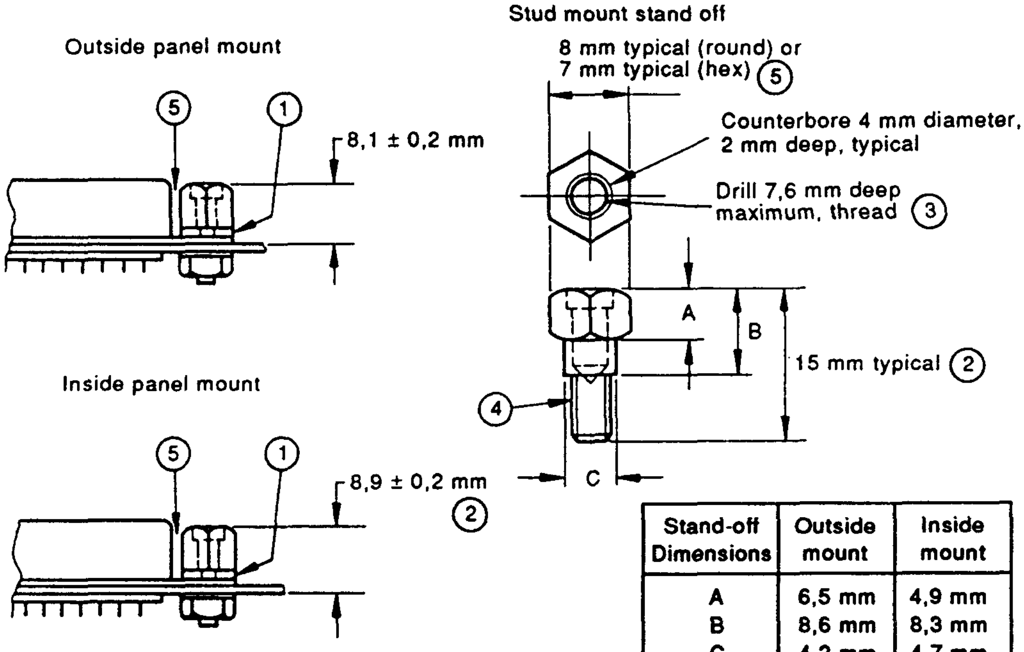


Figure 18 – Connector panel cut-out

28.3 The connector shall be attached to the device with one of the stud mount stand-offs shown in figure 19 as determined by the panel mounting method used.



Stand-off Dimensions	Outside mount	Inside mount
A	6,5 mm	4,9 mm
B	8,6 mm	8,3 mm
C	4,2 mm	4,7 mm

1 Split lock washer, 1,2 mm typical thickness

2 Typical dimension, determined by panel and lock washer thickness, connector housing

3 ISO metric thread M3,5 x 0,6

4 Thread size designer's choice (does not affect compatibility)

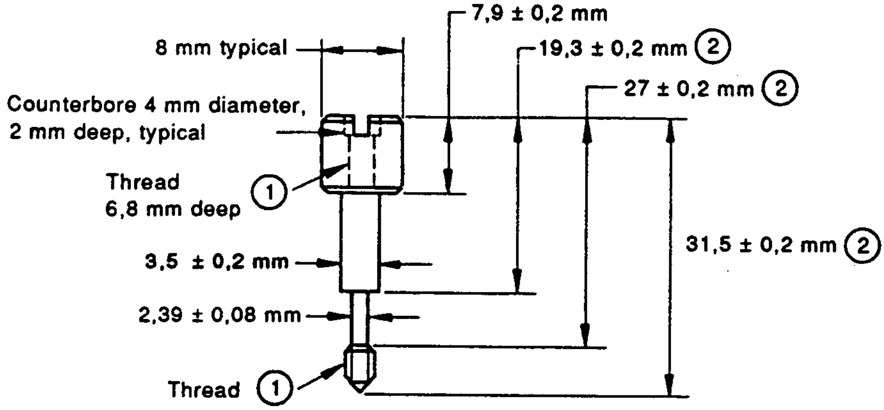
5 Care should be taken to allow clearance for mating shell connector

Figure 19 – Stud mount dimensions

29 Cable assembly

29.1 The cable assembly shall be provided with both a plug and a receptacle connector type at each end of the cable. The preferred method of assembling the stacked connectors contains a rigid structure (assures a reliable and positive connection of multiple cable assemblies) as shown in figure 21.

29.2 Each connector assembly shall be fitted with a pair of captive locking screws. Each lock screw shall conform to the mechanical dimension shown in figure 20. A retaining ring, or equivalent, shall be used to retain the lock screw as a captive element.



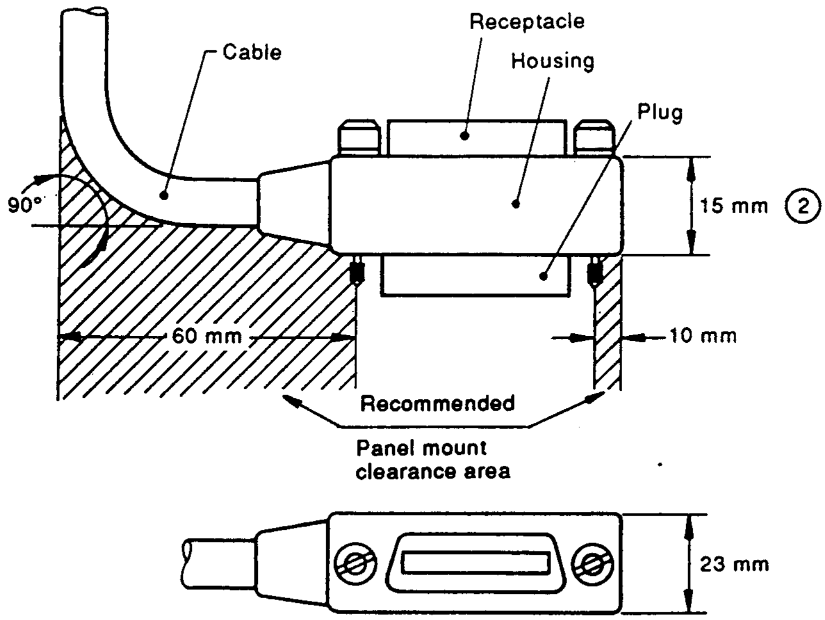
NOTES

(1) ISO metric thread M3,5 x 0,6

(2) Length is a function of housing dimension; length values shown apply to 15 mm typical housing height (see figure 21)

Figure 20 – Locking screw

29.3 It is recommended that each pair of connectors, assembled according to 29.1, be partially enclosed within a suitable housing as shown in figure 21. Individual cable assemblies may be of any length up to 4 m. The housing may be of plastic or metallic matter, the latter is preferred for superior EMC performance; see annex J for additional information on appropriate means for screening the complete cable assembly.



NOTES

1 All measurements are typical.

2 Length of lock screw is a function of this dimension.

Figure 21 – Cable connector housing

SECTION 5: SYSTEM APPLICATIONS AND GUIDELINES FOR THE DESIGNER

30 System compatibility

30.1 *General guidelines*

This interface system offers a wide range of capability from which to choose the appropriate interface functions to fit different applica'ions. Within most interface functions a number of options are available. In addition, the designer has freedom to select all the device-dependent capabilities contained within the device functions.

30.1.1. It is the responsibility of the designer to define the complete capability of a device (interface system choices and related device-dependent interactions) so that the end user of the device can efficiently interface and programme the device for appropriate system applications.

30.1.2 Selection of a minimum set of interface functions from section 2 leads to the following minimum set of signal lines in order to be system compatible:

- DIO 1 to 7;
- DAV, NRFD, NDAC;
- IFC and ATN (unnecessary in systems without a controller).

30.1.3 In order to provide system compatibility a designer shall not introduce new interface functions beyond those defined in section 2.

31 Data rate consideration

Designers of devices intended to communicate over the interface system bus are advised to consider the relationships between various levels of system performance and the specific device circuits used to provide these different levels of performance. The following statements are intended as guidance.

31.1 The interface bus will operate at distances up to 20 m at a maximum of 250 000 bytes per second, with an equivalent standard load for each 2 m of cable using 48 mA open collector drivers.

31.2 The interface bus will also operate at 20 m, 500 000 bytes per second, with an equivalent standard load for each 2 m of cable using 48 mA three-state drivers.

31.3 *Higher speed operation*

To achieve the maximum possible data transfer rate (normally up to one million bytes per second) within a system the designer should:

- use 48 mA three-state drivers;
- use total cable lengths as short as possible up to a maximum of 15 m with at least one equivalent load for each metre of cable;
- ensure that all devices are powered on;
- minimize device capacitance on each lead (REN and IFC excepted) to less than 50 pF per device;
- use a minimum T_1 value of 350 ns;
- add, as required, multiple resistive (powered) loads, beyond one per signal line per device, up to 15 loads per interconnected system.

Warning: Satisfactory operation (i.e. data transfer without error) is not assured when devices designed with the minimum T_1 value of 350 ns are operated under conditions other than those specified above.

NOTES

1 Devices with a T_1 value of < 700 ns, device capacitance of 50 pF, or having multiple resistive loads shall be so marked, as acceptable variants.

2 Use of a data byte buffer store within the device may be advantageous.

32 Device capabilities

32.1 *Busy function*

In system operation it is useful to either programme a device or initiate some operation within a device and then proceed to communicate with other devices (while the first device is busy carrying out the required task). The busy (operation being completed) function is a device state and not an interface state. In order to permit interface bus communication independent of the busy condition of a device, three possible methods are available:

- NRFD hold;
- SRQ and serial poll;
- parallel poll.

Both the serial poll and parallel poll methods are described in section 2.

32.1.1 *NRFD hold*

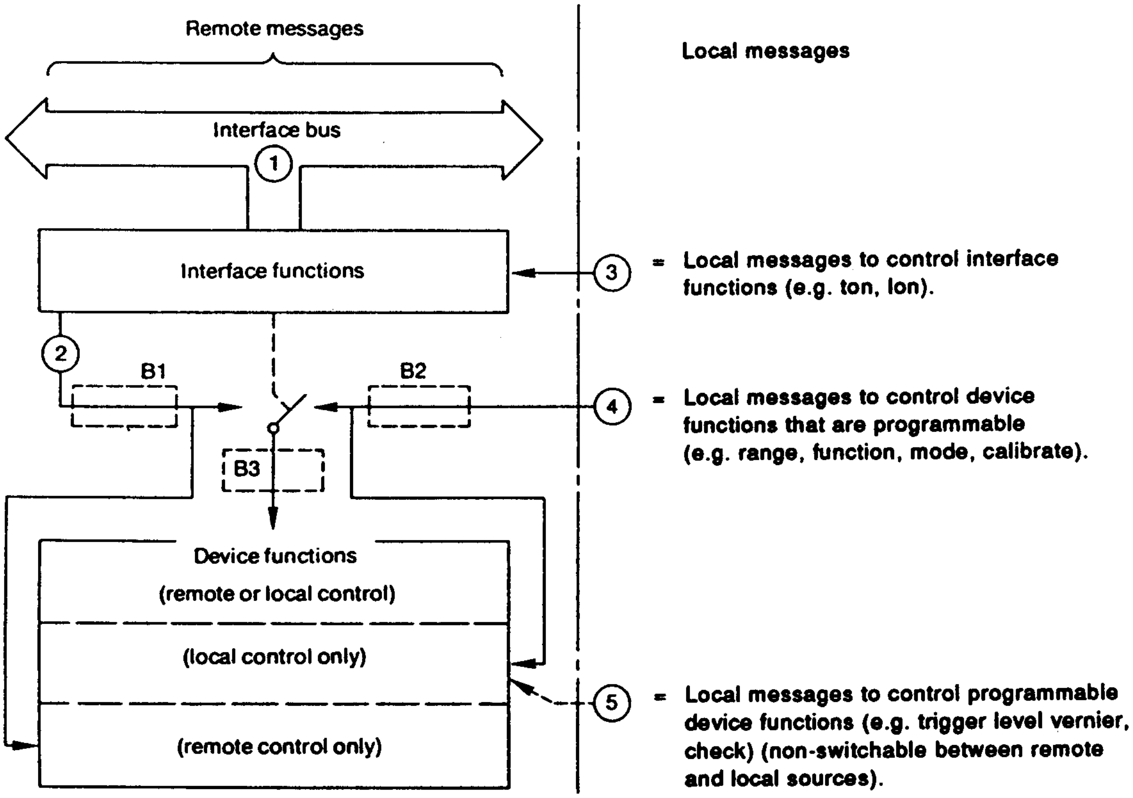
The NRFD signal line may be gated to incorporate the busy function. In so doing, the NRFD signal (or RFD message) *changes* its definition to include more than the normal "not ready for the next data byte" meaning. The internal busy signal is gated to the NRFD signal line *through* the AH function. In this manner the device may be unaddressed as a listener during a "busy cycle" and the interface bus may be used for other purposes. When readdressed as a listener, the device will indicate its internal busy status to the interface. The device indicates "busy" by setting NRFD to 1 and indicates "operation complete" by setting NRFD to Ø.

> *Caution:* If NRFD hold is used for the busy function where a device may not recover or may never reach the non-busy condition, then another listen address (always accessible) should be available to clear the potential hang-up condition.

32.2 *Remote local applications*

32.2.1 The designer is free to implement, within a device, whatever programmable *device functions* are appropriate for a particular device application(s). The designer is not free to programme remotely the local control functions which interact directly with the *interface functions* as specified throughout section 2.

32.2.2 To implement a programmable device capable of being controlled either remotely or locally may require the switching of some or all of the typical controls illustrated in figure 22. This figure is not meant to imply a comprehensive set of switching techniques, switching locations, or switched message contents.



1 Remote messages (e.g. ATN, MLA interface messages, DAB device messages).

2 Remote control of programmable device functions (e.g. range, function) and the execution of device functions (e.g. clearing, triggering).

B Indicates possible locations of buffer storage elements. Normally B1 and B2 are used in combination or B3 is used alone.

Figure 22 – Remote/local message paths

33 AND and OR logic operations

The message sent by one interface fuction is not necessarily the same message received by another interface function (irrespective of time differences due to transmission characteristics of the signal lines) in the case of three messages as used in the SH, AH, and SR interface functions:

- the RFD (or DAC) message received (by an SH function) must be the logical AND of all RFD (or DAC) messages sent (by all AH functions);
- the SRQ message received (by a C function) must be the logical OR of all SRQ messages sent (by all SR functions).

NOTE - The DAV message received (by all AH functions) must be the DAV message sent (by one and only one SH function).

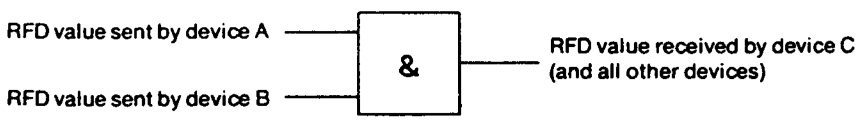
33.1 *RFD and DAC messages*

33.1.1 The RFD (or DAC) message sent true (or false) respectively by an AH function is performed by setting the NRFD (or NDAC) signal line to Ø (high) or driving the NRFD (or NDAC) signal line to 1 (low) respectively.

33.1.2 The RFD (or DAC) message received by an SH is received true when the state of the signal line is Ø (high) which means that all RFD (or DAC) messages sent are passive true.

33.1.3 The RFD (or DAC) message received by an SH is received false when the state of the signal line is 1 (low) which means that one or more RFD (or DAC) messages sent are false.

33.1.4 The logical equivalent of these conditions is illustrated below.



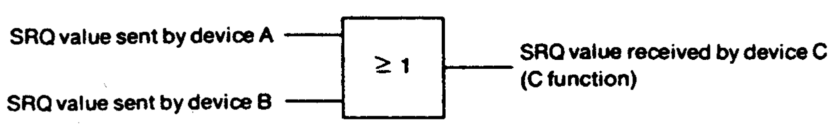
33.2 *SRQ message*

33.2.1 The SRQ message sent true or false by an SR function is performed by driving the SRQ signal line to 1 (low) or setting the SRQ signal line to Ø (high) respectively.

33.2.2 The SRQ message received by the C function is received true when the state of the bus signal line is 1 (low) which means that one or more SR functions have sent the SRQ message true.

33.2.3 The SRQ message received by a controller function is received false when the state of the bus signal line is Ø (high) which means that all SR functions have sent the SRQ message passive false.

33.2.4 The logical equivalent of these conditions is illustrated below:



33.3 *Circuit implementations*

33.3.1 A typical circuit configuration with which these functions on the respective bus signal lines can be performed is that represented in clause 21, figure 16. The driver element must be a bi-state (open collector) driver as represented in figure 23.

NOTE - Whether or not invertors are used to convert the internal representation of the RFD (or DAC) message into the actual message sent on the bus signal lines depends on the internal assertion definition for true and false with respect to the "high" or "low" voltage levels used internal to the device.

This matter is left to the designer.

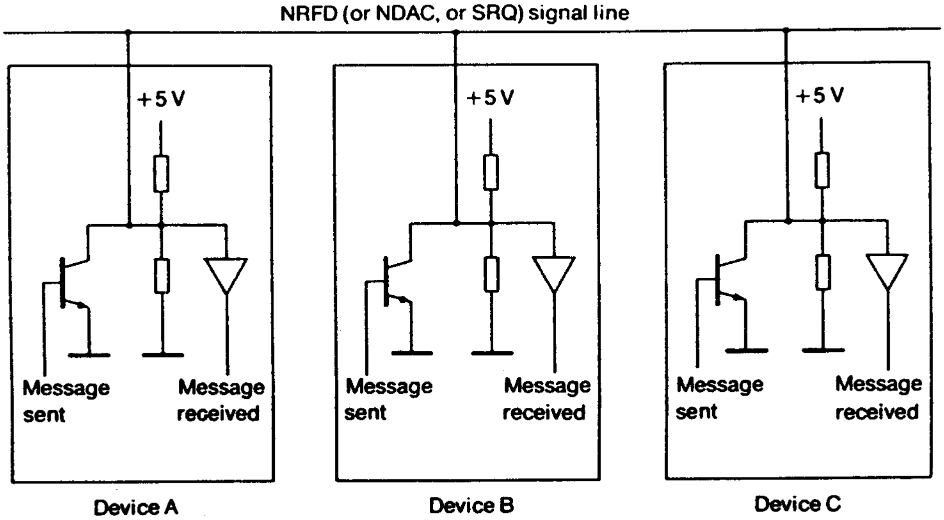


Figure 23 – Bi-state signal line logic (open collector drivers)

33.3.2 Typical signals presented to the NRFD (or SRQ) interface bus signal lines by devices A and B as described in 33.1 and 33.2 may be represented as shown in figure 24. Only the *composite* signal line waveform as received at device C exists on the bus. The signal levels shown for devices A and B exist only within the devices' drivers and not on the bus signal line.

34 Address assignment

34.1 Normally, a device will be assigned a single talk and single listen address to perform the essential tasks. It may be useful to design a device with multiple talk (or listen) addresses to facilitate system requirements. A device could be assigned two talk addresses (e.g. one to output raw data, the other to output processed data). Care should be taken to minimize the use of such multiple addresses as later system configurations may be restricted due to excessive use of primary addressing capability.

35 Typical combinations of interface functions

The designer is free to select the particular interface functions required to meet specific device applications. The selection of certain interface functions requires the inclusion of other interface functions as defined throughout the allowable subset clauses of section 2. The list below represents typical combinations of interface functions and does not imply that these are the only combinations possible or useful.

Device	Typical interface functions used
Signal generator (only able to listen)	AH, L, RL, DT
Tape reader (only able to talk)	SH, AH, T
Digital voltmeter (able to talk and listen)	SH, AH, T, L, SR, RL, PP, DC, DT
Calculator (able to talk, listen, and control)	SH, AH, T, L, C

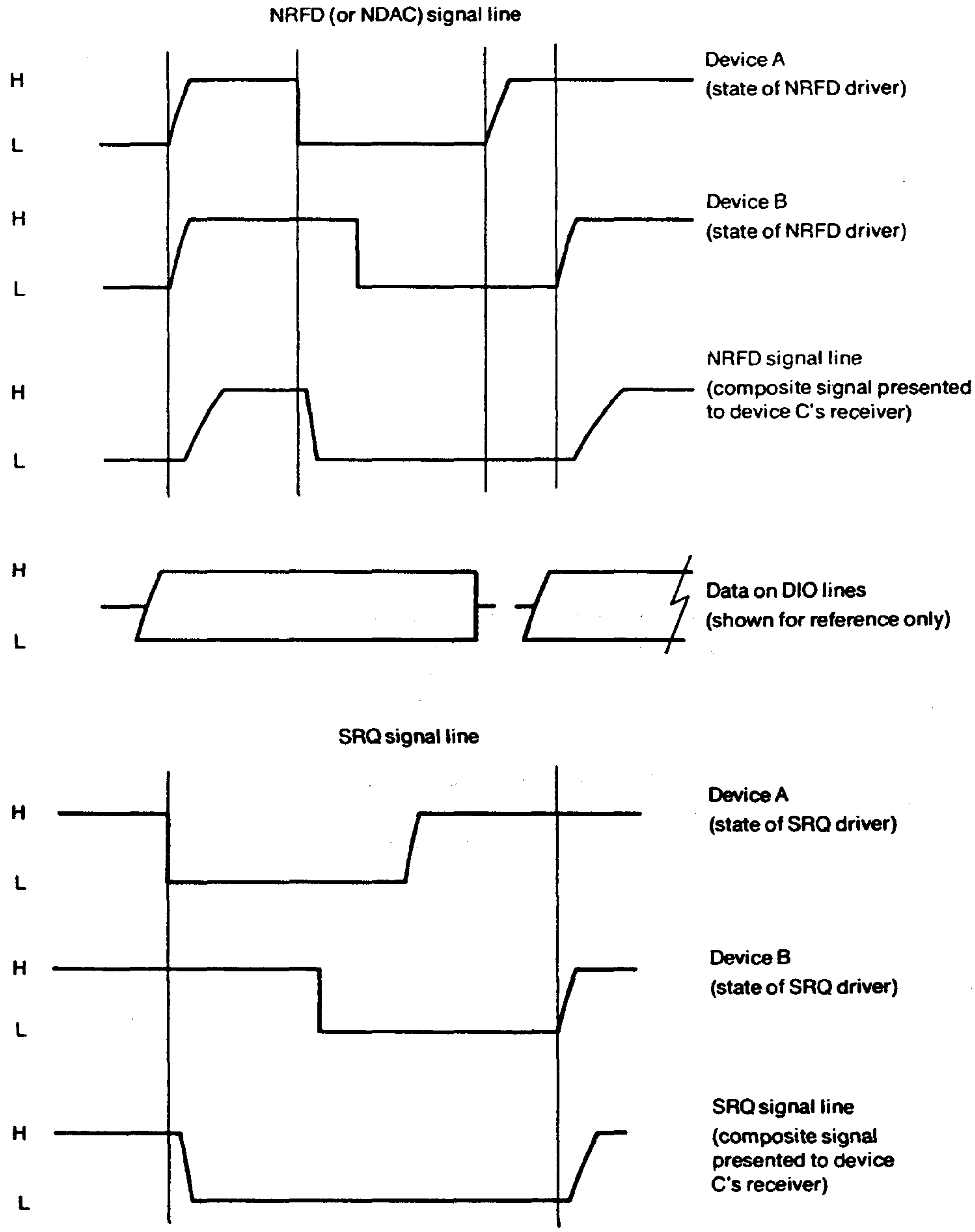


Figure 24 – Signal line logic and timing relationships

36 Unimplemented interface message handling

When the ATN message is true, a device should ignore all multiline messages that are inappropriate given the current states of the implemented interface functions. A device shall handshake the inappropriate multiline message, but should not take further action including recording an error, requesting service, or interrupting the exchange of remote messages. Subsequent messages should be processed in the normal manner.

EXAMPLES

1 If a device implements the DT0 subset, the group execute trigger (GET) remote message should be ignored by the device.

2 If the LE interface function is in LPIS and the PP interface function is in PUCS, the my secondary address (MSA) remote message should be ignored by the device.

3 If an addressed command group remote message that is not GTL, SDC, PPC, GET, or TCT or a universal command group remote message that is not LLO, DCL, PPU, SPE, or SPD is received, the device should ignore the message.

SECTION 6: SYSTEM REQUIREMENTS AND GUIDELINES FOR THE USER

37 System compatibility

37.1 *General guidelines*

Devices designed to this interface system may have a wide range of capability relative to their ability to communicate over the interface. This part does *not* cover the *operational* characteristics of *devices*, only the mechanical, electrical, and functional capabilities of the interface system.

37.1.1 *The burden of responsibility for system compatibility at the operational level is on the user.* The user shall be familiar with all device characteristics interacting with the interface system (e.g. device-dependent programme codes, output data format and codes, etc.).

38 System installation requirements

38.1 *System configuration restrictions*

38.1.1 *Maximum number of devices*

The maximum number of devices that can be connected together to form one interface system is 15.

38.1.2 *Minimum system configurations*

38.1.2.1 An interface system shall contain one or more devices containing at least one Talker function, one Listener function, and one Controller function.

38.1.2.2 If all the Talker functions include the use of the *ton* message (Talker types T1, T3, T5, T7, TE1, TE3, TE5, or TE7), and all the Listener functions include the *lon* message (Listener types L1, L3, LE1 or LE3), a system may be operated without a Controller function when the *ton* and *lon* messages are true. The *lon* and *ton* messages are normally provided by local switches.

38.1.3 *System controllers*

All system configurations containing more than one controller shall satisfy the following conditions:

- there shall not be more than one controller function in a system that is in the SYSTEM CONTROL ACTIVE STATE (SACS);
- every controller in the system shall be able to pass and receive control of the interface.

38.1.4 *Devices powered off and on*

38.1.4.1 A system will operate without adversely affecting normal data transfer with at least two-thirds of the devices powered on. A system will operate correctly with any number of devices powered off, provided all those devices powered off do not degrade the specified high state condition (i.e. the voltage on each signal line with all its output drivers passive false should exceed +2,5 V with respect to the logic ground at each device).

38.1.4.2 Unless special precautions are taken (i.e. use of special driver circuits beyond the scope of this part) powering a device on while the system is running may cause faulty operation.

39 Address assignment

39.1 *Primary talk addresses*

39.1.1 A device that contains a Talker function or an Extended Talker function may be assigned any value for bits T1 to T5 of its MY TALK ADDRESS (MTA) message code other than:

T5	T4	T3	T2	T1
1	1	1	1	1

NOTE - This restriction is maintained only for compatibility with earlier systems (identified as UNT).

39.1.2 Two or more Talker functions (whether within the same or separate devices) shall not be assigned the same value for bits T1 to T5 of their MTA codes.

39.1.3 A device that contains a Talker and Listener function may be assigned a Talk address such that T1 to T5 of its MTA code equals L1 to L5 of its MLA code.

39.1.4 An Extended Talker Interface function shall not be assigned the same value for bits T1 to T5 of its MTA code as that assigned to a Talker function.

39.2 *Primary listen addresses*

39.2.1 A device that contains a Listener function or an Extended Listener function may be assigned any value for bits L1 to L5 of its MY LISTEN ADDRESS (MLA) code other than:

L5	L4	L3	L2	L1
1	1	1	1	1

39.2.2 Two or more Listener functions (usually within separate devices) may be assigned the same value for bits L1 to L5 of their MLA codes.

39.2.3 A device that contains both a Listener and Talker function may be assigned a Listen address such that L1 to L5 of its MLA code equals T1 to T5 of its MTA code.

39.3 *Secondary addresses*

39.3.1 A device that contains an Extended Talker function or Extended Listener function may be assigned any value for bits S1 to S5 of its MY SECONDARY ADDRESS (MSA) code other than:

S5	S4	S3	S2	S1
1	1	1	1	1

39.3.2 Two or more Extended Talker functions (whether within the same or separate devices) shall not be assigned the same value for both T1 to T5 of their MTA codes and bits S1 to S5 of their MSA codes.

39.3.3 Two or more Extended Listener functions (usually within separate devices) may be assigned the same value for both bits L1 to L5 of their MLA codes and bits S1 to S5 of their MSA codes.

39.3.4 A device that contains both an Extended Talker and Extended Listener function may be assigned a Listen address such that L1 to L5 of its MLA code equals T1 to T5 of its MTA code and both functions may utilize the same secondary address.

40 Cabling restrictions

40.1 *Maximum cable length*

The maximum length of cable that may be used to connect together a group of devices within one bus system is:

- 2 m times the number of devices;
- or 20 m, whichever is less.

40.2 *Distribution of maximum cable length*

The maximum length of cable as defined in 40.1 may be distributed among the devices in a system in any manner deemed suitable by the user. Caution should be taken if any individual cable length exceeds 4 m.

40.3 *Cabling configurations*

Cables may be interconnected in any manner deemed suitable by the user (i.e. star, linear, or combinations thereof).

40.4 *Ground requirement precaution*

Devices should not be operated at significantly different frame potentials as the system may not be capable of handling excessive ground currents.

41 Operational sequences

Most interface communication tasks require a sequence of coded messages to be sent over the interface. The following sequences are recommended for the tasks specified, although there are many others which might be found useful.

NOTE - [...] indicates optional segments of a sequence.
(...) indicates messages not uniquely defined in this part.

41.1 *Data transfer*

ATN		
[1	UNL]	– inhibits all current listeners (can be omitted if not required);
1	$(LAD)_1$	– each address sent enables a specific device to receive future data bytes;
[1	$(LAD)_n$]	– more than one address may be sent if multiple listeners are desired;
1	(TAD)	– the address sent enables a specific device to send data as soon as ATN becomes Ø;
Ø	$(DAB)_1$	– sent by the currently enabled talker to all currently enabled listeners;
[Ø	$(DAB)_n$]	– bytes may be sent until the controller again sets ATN to 1 to repeat the sequence. If the talker is sending a specific-length record, it may optionally set EOI = 1 while sending the last byte or subsequently send the EOS message.

NOTE - (LAD) represents a listen address of a specific device.
(TAD) represents a talk address of a specific device.
(DAB) represents any data byte.

41.2 *Serial poll* (issued by controller usually whenever SRQ = 1 on the interface)

ATN		
1	UNL	– prevents other devices from listening to status sent (controller continues to listen without being addressed);
1	SPE	– puts interface into Serial Poll Mode during which all devices send status instead of data when enabled;
1	$(TAD)_n$	– enables a specific device to send status. Within this loop, devices should be sequentially enabled;
Ø	(SBN) or (SBA)	– status byte sent by enabled device. If SBN was sent, loop should be repeated. If SBA was sent, the enabled device is identified as having sent SRQ over the interface and will automatically remove it;
	or	
1	SPD	– removes the interface from serial poll mode;
1	[UNT]	– disables last talker from sending data if ATN is set Ø.

NOTE - (TAD) represents a talk address of a specific device.
(SBN) represents a composite status byte sent by a device in which a request for service is not indicated (bit 7 = Ø) (SBN = STB ∧ $\overline{RQS}$).
(SBA) represents a composite status byte sent by a device in which a request for service is indicated (bit 7 = 1) (SBA = STB ∧ RQS).

41.3 *Control passing*

ATN		
1	(TAD)	– the address sent should be that of the device to which control is being passed;
1	TCT	– notifies addressed device to take over control of the interface;
∅	–	– new controller-in-charge at this time.

NOTE - (TAD) represents a talk address of a specific device.

41.4 *Parallel poll*

41.4.1 *Parallel poll configure*

ATN	IDY		
1	∅	(LAD)	– addresses a particular device for which a parallel response coding is to be assigned;
1	∅	PPC	– enables the addressed listener to be configured;
1	∅	PPE	– bit 4 specifies the sense of the poll response, bits 1 ... 3 specify, in binary code, the DIO line on which the poll response is to be given;
1	∅	UNL	– end of the configuration routine.

NOTES

1 (LAD) represents a listen address of a specific device.

2 The PPE command can be cleared by a PPD command.

3 The configuration can be cleared by a PPU command.

41.4.2 *Parallel poll response*

ATN	IDY	
1	1	– whenever the bus is in this state, predetermined devices will each place their requests on a specific DIO line. If more than one device is sharing a DIO line, the line value can indicate either an OR or an AND conjunction of requests depending on commands previously sent to the devices instructing them to use the ∅ or 1 value to request service.

41.5 *Placing devices in forced remote control*

ATN	REN		
1	1	LLO	– disables all devices' "return-to-local" button;
1	1	$(LAD)_i$	– each address sent places the addressed device into remote
1	1	·	state, disabling all local controls.
1	1	·	
1	1	$(LAD)_n$	

NOTES

1 (LAD) represents a listen address of a specific device. (Devices will all revert back to local state as a group at any time a ∅ value of REN is placed on the interface.)

2 Selected local controls may be re-enabled by sending device-dependent remote messages.

41.6 *Sending interface clear*

While the IFC message is being sent only the DCL, LLO, PPU and REN universal commands will be recognized.

41.7 *Restriction on the use of* tca

The designer shall not assume that valid data will be transferred across the interface if the *tca* message becomes true while a device-dependent message is true. See 15.5.1.2 for background.

Annex A
(informative)

Typical instrument system

The typical system shown in figure A.1 illustrates the capability of the interface system to handle a variety of instrumentation system needs.

Two possible event sequences, to accomplish specific measurement tasks using the interface system, are included as examples listed below.

Sequence No. 1 – Processor programmes instruments and initiates measurements; resulting device-dependent data are returned to processor.

Sequence No. 2 – Processor programmes instruments and initiates measurements; resulting device-dependent data are returned to another device.

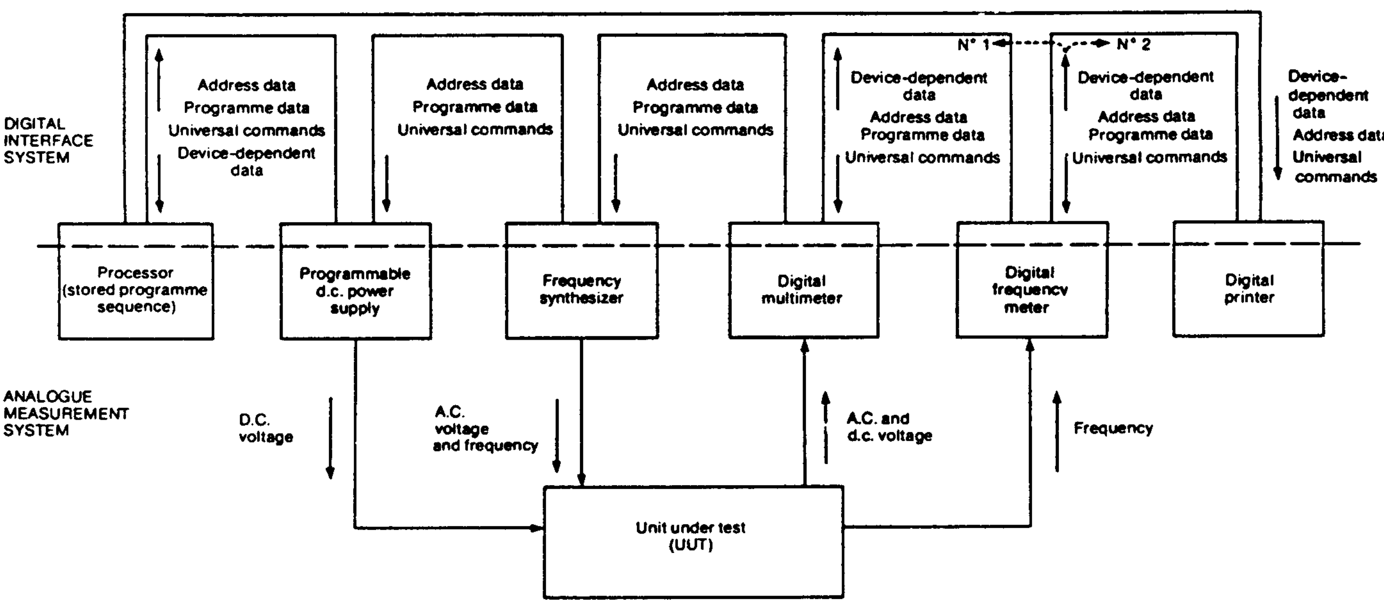


Figure A.1 – Typical instrument system

Event sequence No. 1 (device-dependent data returned to processor)

1) Processor initializes the interface system by sending the IFC message true.

2) Processor causes all devices to set their internal conditions to a predefined state by sending the DCL message true.

3) Processor sends the Listen address of the d.c. power supply followed by programme data for that device.

4) Processor sends the unlisten command, then the Listen address for the next device, followed by programme data for it.

5) Event No. 4 is repeated until each device of interest for this specific test has been addressed and programmed, then the unlisten command is sent.

6) Processor sends Listen address of selected measurement device (e.g. the digital frequency meter), then that program code required to initiate a measurement.

7) Processor sends unlisten command, addresses itself to Listen, then sends talk address of the measurement device.

8) Upon completion of its internal measurement cycle, the digital frequency meter sends (talks) its measurement results (device-dependent data) to the addressed Listener, the processor.

Event sequence No. 2 (device-dependent data directed to digital printer)

1-6) Are identical to event sequence No. 1.

7) Processor sends unlisten command, then the Listen address of the Digital Recorder, followed by the Talk address of the measurement device.

8) Upon completion of its measurement, the measurement device again sends its resulting device-dependent data to the addressed Listener, the Digital Recorder.

NOTE - If the processor were to address *both* the digital recorder and itself, the resulting device-dependent data would be accepted by both devices, even though the two may have vastly different rates at which data can be accepted.

Annex B
(informative)

Handshake process timing sequence

B.1 General comments

Each data byte transferred by the interface system uses the handshake process to exchange data between source and acceptor. Typically, the source is a Talker and the acceptor a Listener.

Figure B.1 illustrates the handshake process by indicating the actual waveforms on the DAV, NRFD, and NDAC signal lines. The NRFD and NDAC signals each represent composite waveforms resulting from two or more Listeners accepting the same data byte at slightly different times due to variations in the transmission path length and different response rates (delays) to accept and process the data byte.

Figure B.2 represents the same sequence of events, in flow chart form, to transfer a data byte between source and acceptor.

The annotation numbers on the flow chart and the timing sequence diagram refer to the same event on the list of events.

B.2 Handshake process timing diagram

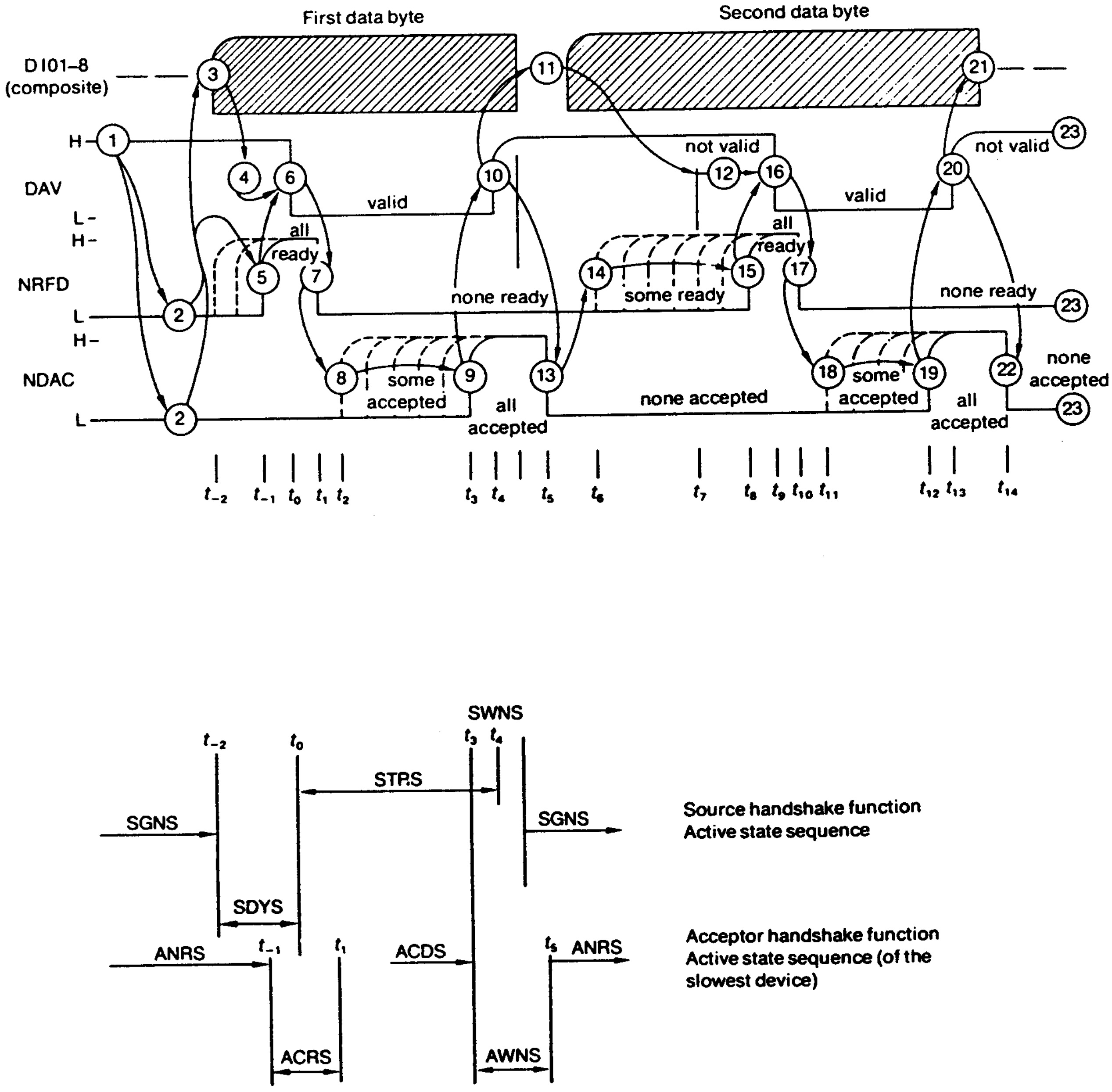


Figure B.1 – Signal line timing sequence for one talker and multiple listeners using the handshake process.

Numbered events are referenced in clauses B.3 and B.4

NOTE - H ≥ +2,0 V, L ≤ +0,8 V.

B.3 Handshake process flow diagram

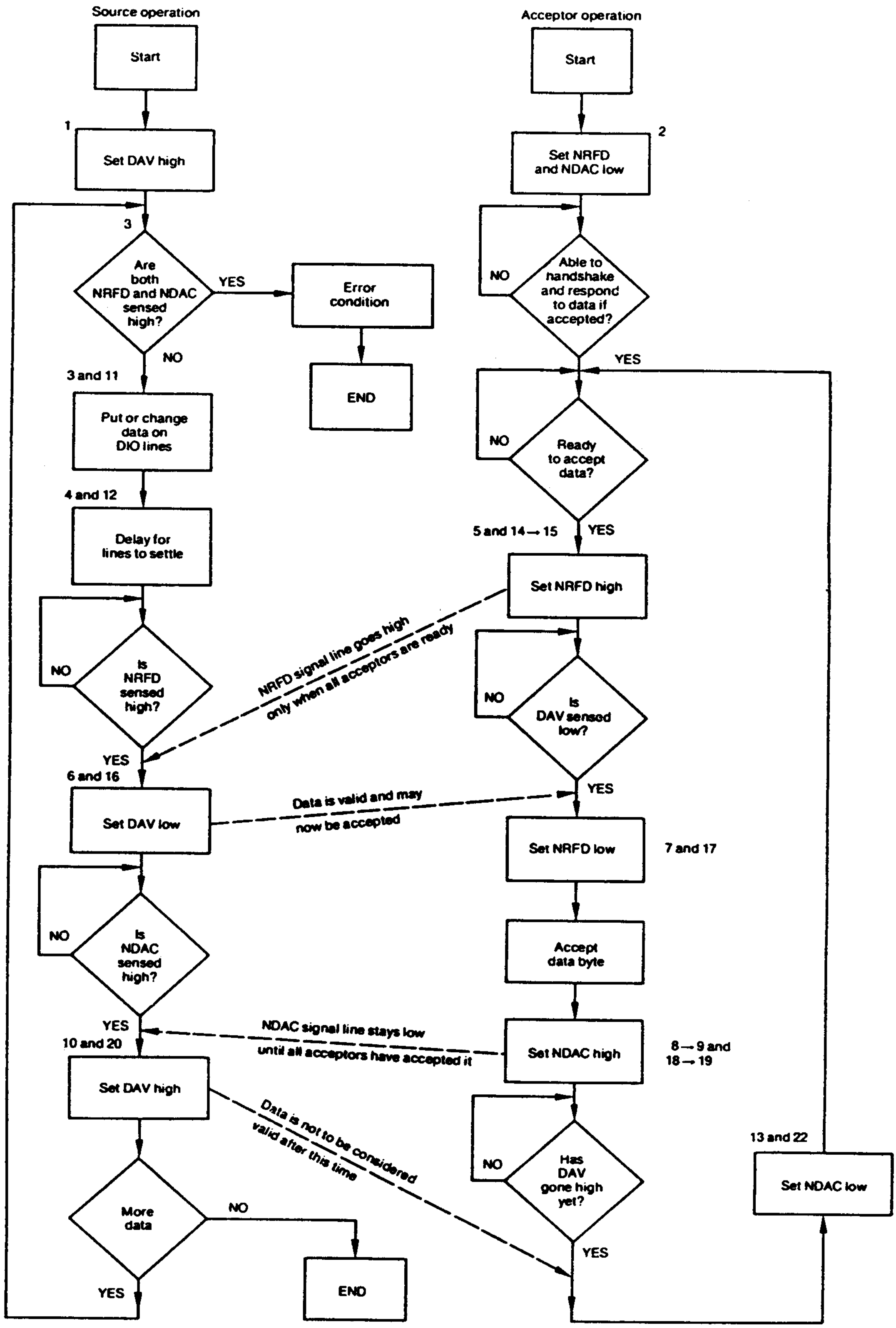


Figure B.2 – Logical flow of events for source and acceptor when transferring data using the handshake process (see list of events on next page)

NOTE - This flow diagram is not intended to represent the only method of implementing an acceptor handshake (see 7.5.3).

B.4 List of events for handshake process

1	–	Source initializes DAV to High (data not valid).
2	–	Acceptors initialize NRFD to Low (none are ready for data), and set NDAC to Low (none have accepted the data).
3	t_{-2}	Source checks for error condition (both NRFD and NDAC High), then sets data byte on DIO lines.
4	$t_{-2} \rightarrow t_0$	Source delays to allow data to settle on DIO lines.
5	t_{-1}	Acceptors have all indicated readiness to accept first data byte; NRFD line goes High.
6	t_0	Source, upon sensing NRFD High, sets DAV Low to indicate that data on DIO lines are settled and valid.
7	t_1	First acceptor sets NRFD Low to indicate that it is no longer ready, then accepts the data. Other acceptors follow at their own rates.
8	t_2	First acceptor sets NDAC High to indicate that it has accepted the data. (NDAC remains Low due to other acceptors driving NDAC Low.)
9	t_3	Last acceptor sets NDAC High to indicate that it has accepted the data; all have now accepted and the NDAC line goes High.
10	t_4	Source, having sensed that NDAC is High, sets DAV High. This indicates to the acceptors that data on the DIO lines must now be considered not valid.
11	t_4-t_7	Source changes data on the DIO lines.
12	t_7-t_9	Source delays to allow data to settle on DIO lines.
13	t_5	Acceptors, upon sensing DAV High (at 10) set NDAC Low in preparation for next cycle. NDAC line goes Low as the first acceptor sets the line Low.
14	t_6	First acceptor indicates that it is ready for the next data byte by setting NRFD High. (NRFD remains Low due to other acceptors driving NRFD Low.)
15	t_8	Last acceptor indicates that it is ready for the next data byte by setting NRFD High; NRFD signal line goes High.
16	t_9	Source, upon sensing NRFD High, sets DAV Low to indicate that data on DIO lines are settled and valid.
17	t_{10}	First acceptor sets NRFD Low to indicate that it is no longer ready, then accepts the data.
18	t_{11}	First acceptor sets NDAC High to indicate that it has accepted the data (as in 8).
19	t_{12}	Last acceptor sets NDAC High to indicate that it has accepted the data (as in 9).
20	t_{13}	Source, having sensed that NDAC is High, sets DAV High (as in 10).
21	–	Source removes data byte from DIO signal lines after setting DAV high.
22	t_{14}	Acceptors, upon sensing DAV High, set NDAC Low in preparation for next cycle.
23	–	Note that all three handshake lines are at their initialized states, as at 1 and 2.

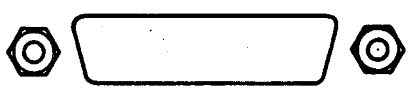
Annex C
(informative)

Interface function allowable subset reference list

Capability identification code

It may be useful to place an IEC 625-1 capability code below or near the interface connector on each device to identify the complete set of interface functions contained within that device. In this part each interface function and allowable subset thereof has an equivalent alphanumeric code to identify that capability. All such interface function capability codes may be expressed in a concise alphanumeric string and marked on the exterior of the device to facilitate user system assembly.

For example, a device with the basic talker function, the ability to send status bytes, the basic listener function, a listen only mode switch, service request capability, remote local capability without local lockout, manual configuration of the parallel poll capability, complete device clear capability, no capability for device trigger, and no controller capability would be identified with the following code:



SH1, AH1, T2, L1, SR1, RL2, PP2, DC1, DT0, C0, E1

It is permissible to indicate the absence of capability with the 0 symbol (e.g. DT0 means no device trigger capability). In the above example, the eight interface functions are identified. In addition, the type of electrical interface contained within the device is specified. The notation E1 is used to identify open collector drivers and E2 is used to identify three-state drivers.

Any further device capability, useful for system configuration, may be added at the appropriate place on the physical equipment and in the relevant documentation for that equipment.

Source handshake function allowable subsets

Identification	Description	States omitted	Other requirements	Other function subsets required
SH0	No capability	All	None	None
SH1	Complete capability	None	None	T1-T8 or TE1-TE8 or C5-C28

Acceptor handshake function allowable subsets

Identification	Description	States omitted	Other requirements	Other function subsets required
AHØ	No capability	All	None	None
AH1	Complete capability	None	None	None

Talker function allowable subsets

Identi-fication	Description					States omitted	Other requirements	Other function subsets required
	Capabilities:							
	Basic talker	Serial poll	Talk only mode	Unaddress if MLA				
TØ	N	N	N	N		All	None	None
T1	Y	Y	Y	N		None	Omit [MLA ∧ (ACDS)]	SH1 and AH1
T2	Y	Y	N	N		None	1. Omit [MLA ∧ (ACDS)] 2. ton always false	SH1 and AH1
T3	Y	N	Y	N		SPIS, SPMS and SPAS	Omit [MLA ∧ (ACDS)]	SH1 and AH1
T4	Y	N	N	N		SPIS, SPMS and SPAS	1. Omit [MLA ∧ (ACDS)] 2. ton always false	SH1 and AH1
T5	Y	Y	Y	Y		None	Include [MLA ∧ (ACDS)]	SH1 and L1-L4 or LE1-LE4
T6	Y	Y	N	Y		None	1. Include [MLA ∧ (ACDS)] 2. ton always false	SH1 and L1-L4 or LE1-LE4
T7	Y	N	Y	Y		SPIS, SPMS and SPAS	Include [MLA ∧ (ACDS)]	SH1 and L1-L4 or LE1-LE4
T8	Y	N	N	Y		SPIS, SPMS and SPAS	1. Include [MLA ∧ (ACDS)] 2. ton always false	SH1 and L1-L4 or LE1-LE4

Talker function (with address extension) allowable subsets

Identi-fication	Description					States omitted	Other requirements	Other function subsets required
	Capabilities:							
	Basic extended talker	Serial poll	Talk only mode	Unaddress if MSA ∧ (LPAS)				
TEØ	N	N	N	N		All	None	None
TE1	Y	Y	Y	N		None	Omit [MSA ∧ (LPAS) ∧ (ACDS)]	SH1 and AH1
TE2	Y	Y	N	N		None	1. Omit [MSA ∧ (LPAS) ∧ (ACDS)] 2. ton always false	SH1 and AH1
TE3	Y	N	Y	N		SPIS, SPMS and SPAS	Omit [MSA ∧ (LPAS) ∧ (ACDS)]	SH1 and AH1
TE4	Y	N	N	N		SPIS, SPMS and SPAS	1. Omit [MSA ∧ (LPAS) ∧ (ACDS)] 2. ton always false	SH1 and AH1
TE5	Y	Y	Y	Y		None	Include [MSA ∧ (LPAS) ∧ (ACDS)]	SH1 and L1-L4 or LE1-LE4
TE6	Y	Y	N	Y		None	1. Include [MSA ∧ (LPAS) ∧ (ACDS)] 2. ton always false	SH1 and L1-L4 or LE1-LE4
TE7	Y	N	Y	Y		SPIS, SPMS and SPAS	Include [MSA ∧ (LPAS) ∧ (ACDS)]	SH1 and L1-L4 or LE1-LE4
TE8	Y	N	N	Y		SPIS, SPMS and SPAS	1. Include [MSA ∧ (LPAS) ∧ (ACDS)] 2. ton always false	SH1 and L1-L4 or LE1-LE4

Listener function allowable subsets

Identification	Description: Capabilities: Basic listener	Listen only mode	Unaddress if MTA		States omitted	Other requirements	Other function subsets required
L∅	N	N	N		All	None	None
L1	Y	Y	N		None	Omit [MTA ∧ (ACDS)]	AH1
L2	Y	N	N		None	1. Omit [MTA ∧ (ACDS)] 2. lon always false	AH1
L3	Y	Y	Y		None	Include [MTA ∧ (ACDS)]	AH1 and T1-T8 or TE1-TE8
L4	Y	N	Y		None	1. Include [MTA ∧ (ACDS)] 2. lon always false	AH1 and T1-T8 or TE1-TE8

Listener function (with address extension) allowable subsets

Identification	Description: Capabilities: Basic extended listener	Listen only mode	Unaddress if [MSA ∧ (TPAS)]*		States omitted	Other requirements	Other function subsets required
LE∅	N	N	N		All	None	None
LE1	Y	Y	N		None	Omit [MSA ∧ (TPAS) ∧ (ACDS)]	AH1
LE2	Y	N	N		None	1. Omit [MSA ∧ (TPAS) ∧ (ACDS)] 2. lon always false	AH1
LE3	Y	Y	Y		None	Include [MSA ∧ (TPAS) ∧ (ACDS)]	AH1 and T1-T8 or TE1-TE8
LE4	Y	N	Y		None	1. Include [MSA ∧ (TPAS) ∧ (ACDS)] 2. lon always false	AH1 and T1-T8 or TE1-TE8

* Replaced by MTA when used together with the T function.

Service request function allowable subsets

Identification	Description	States omitted	Other requirements	Other function subsets required
SRØ	No capability	All	None	None
SR1	Complete capability	None	None	T1, T2, T3, T6 TE1, TE2, TE5 or TE6

Remote local function allowable subsets

Identification	Description	States omitted	Other requirements	Other function subsets required
RLØ	No capability	All	None	None
RL1	Complete capability	None	None	L1-L4 or LE1-LE4
RL2	No local lockout	LWLS and RWLS	rtl always false	L1-L4 or LE1-LE4

Parallel poll function allowable subsets

Identi-fication	Description	States omitted	Other requirements	Other function subsets required
PPØ	No capability	All	None	None
PP1	Remote configuration	None	1. Include [((PPD ∧ PACS) ∨ PPU) ∧ ACDS] 2. Include [PPE ∧ PACS ∧ ACDS] 3. Exclude *lpe*	L1-L4 or LE1-LE4
PP2	Local configuration	PUCS, PACS	1. Include *lpe* 2. Exclude [((PPD ∧ PACS) ∨ PPU) ∧ ACDS] 3. Exclude [PPE ∧ PACS ∧ ACDS] 4. Local messages shall be substituted for S, P1, P2, P3	None

Device clear function allowable subsets

Identification	Description	States omitted	Other requirements	Other function subsets required
DCØ	No capability	All	None	None
DC1	Complete capability	None	None	L1-L4 or LE1-LE4
DC2	Omit selective device clear	None	Omit [SDC ∧ LADS]	AH1

Device trigger function allowable subsets

Identification	Description	States omitted	Other requirements	Other function subsets required
DTØ	No capability	All	None	None
DT1	Complete capability	None	None	L1-L4 or LE1-LE4

Controller function allowable subsets

Identification[3)]	Capabilities										Notes	States required									Other requirements			Other function subsets required				
Note 5	System controller	Send IFC and Take Charge	Send REN	Respond to SRQ	Send I.F. Messages	Receive Control	Pass Control	Pass Control to Self	Parallel Poll	Take Control Synchronously		SNAS, SACS	SIIS, SIAS, SINS	SRIS, SRAS, SRNS	CSNS, CSRS	CACS, CSBS, CSHS, CSWS, CAWS	CADS	CIDS	CTRS	CPWS, CPPS	[TCT ∧ (ACDS) ∧ (TADS)][1)]	[(TADS)][2)]	*tcs* not always false	C1	C2	SH1	AH1, L1-L4 or LE1-LE4	T1-T8, TE1-TE8
C∅	N	N	N	N	N	N	N	N	N	N		O	O	O	O	O	O	O	O	O	O	O	O	O	–	–	–	–
C1	Y	–	–	–	–	–	–	–	–	–	1	R	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–
C2	–	Y	–	–	–	–	–	–	–	–	1	–	R	–	–	–	–	–	–	–	–	–	–	R	–	–	–	–
C3	–	–	Y	–	–	–	–	–	–	–	1	–	–	R	–	–	–	–	–	–	–	–	–	R	–	–	–	–
C4	–	–	–	Y	–	–	–	–	–	–	1	–	–	–	R	–	–	–	–	–	–	–	–	–	–	–	–	–
C5	–	–	–	–	Y	Y	Y	Y	Y	Y	2, 3	–	–	–	–	R	R	R	R	R	R	R	R	–	–	R	R	R
C6	–		–	–	Y	Y	Y	Y	Y	N	2, 3	–	–	–	–	R	R	R	R	R	R	R	O	–	–	R	–	R
C7	–	–	–	–	Y	Y	Y	Y	N	Y	2, 3	–	–	–	–	R	R	R	R	O	R	R	R	–	–	R	R	R
C8	–	–	–	–	Y	Y	Y	Y	N	N	2, 3	–	–	–	–	R	R	R	R	O	R	R	O	–	–	R	–	R
C9	–	–	–	–	Y	Y	Y	N	Y	Y	2, 3	–	–	–	–	R	R	R	R	R	R	O	R	–	–	R	R	R
C10	–	–	–	–	Y	Y	Y	N	Y	N	2, 3	–	–	–	–	R	R	R	R	R	R	O	O	–	–	R	–	R
C11	–	–	–	–	Y	Y	Y	N	N	Y	2, 3	–	–	–	–	R	R	R	R	O	R	O	R	–	–	R	R	R
C12	–	–	–	–	Y	Y	Y	N	N	N	2, 3	–	–	–	–	R	R	R	R	O	R	O	O	–	–	R	–	R
C13	–	–	–	–	Y	Y	N	N	Y	Y	2	–	–	–	–	R	R	R	O	R	O	O	R	–	R	R	R	–
C14	–	–	–	–	Y	Y	N	N	Y	N	2	–	–	–	–	R	R	R	O	R	O	O	O	–	R	R	–	–
C15	–	–	–	–	Y	Y	N	N	N	Y	2	–	–	–	–	R	R	R	O	O	O	O	R	–	R	R	R	–
C16	–	–	–	–	Y	Y	N	N	N	N	2	–	–	–	–	R	R	R	O	O	O	O	O	–	R	R	–	–
C17	–	–	–	–	Y	N	Y	Y	Y	Y	2, 3, 4	–	–	–	–	R	O	R	R	R	R	R	R	–	–	R	R	R
C18	–	–	–	–	Y	N	Y	Y	Y	N	2, 3, 4	–	–	–	–	R	O	R	R	R	R	R	O	–	–	R	–	R
C19	–	–	–	–	Y	N	Y	Y	N	Y	2, 3, 4	–	–	–	–	R	O	R	R	O	R	R	R	–	–	R	R	R
C20	–	–	–	–	Y	N	Y	Y	N	N	2, 3, 4	–	–	–	–	R	O	R	R	O	R	R	O	–	–	R	–	R
C21	–	–	–	–	Y	N	Y	N	Y	Y	2, 3, 4	–	–	–	–	R	O	R	R	R	R	O	R	–	–	R	R	R
C22	–	–	–	–	Y	N	Y	N	Y	N	2, 3, 4	–	–	–	–	R	O	R	R	R	R	O	O	–	–	R	–	R
C23	–	–	–	–	Y	N	Y	N	N	Y	2, 3, 4	–	–	–	–	R	O	R	R	O	R	O	R	–	–	R	R	R
C24	–	–	–	–	Y	N	Y	N	N	N	2, 3, 4	–	–	–	–	R	O	R	R	O	R	O	O	–	–	R	–	R
C25	–	–	–	–	Y	N	N	N	Y	Y	2	–	–	–	–	R	O	O	O	R	O	O	R	–	–	R	R	–
C26	–	–	–	–	Y	N	N	N	Y	N	2	–	–	–	–	R	O	O	O	R	O	O	O	–	–	R	–	–
C27	–	–	–	–	Y	N	N	N	N	Y	2	–	–	–	–	R	O	O	O	O	O	O	R	–	–	R	R	–
C28	–	–	–	–	Y	N	N	N	N	N	2	–	–	–	–	R	O	O	O	O	O	O	O	–	–	R	–	–

O omit
R required
– not applicable or not required
Y yes
N no

1) This is part of the CIDS to CADS transitional expression.
2) This is part of the CACS to CTRS transitional expression.
3) Typical notation to describe a controller consists of the letter C followed by one or more of the numbers indicating the subsets selected. For example: C1, 2, 3, 4, 8.

NOTES

1 One or more of subsets C1 to C4 may be chosen in any combination with any *one* of C5 to C28.

2 Only one subset may be chosen from C5 to C28.

3 The CTRS state shall be included in devices which are to be operated in multi-controller systems.

4 These subsets are not allowed unless C2 is included.

Annex D
(informative)

Interface message reference list

Mnemonic	Message	Interface function(s)
Local messages received (by interface functions)		
gts	go to standby	C
ist	individual service request (qualifier)	PP
lon	listen only	L, LE
[lpe]	local poll enable	PP
ltn	listen	L, LE
lun	local unlisten	L, LE
nba	new byte available	SH
pon	power on	SH, AH, T, TE, L, LE, SR, RL,PP, C
rdy	ready	AH
rpp	request parallel poll	C
rsc	request system control	C
rsv	request service	SR
rtl	return to local	RL
sic	send interface clear	C
sre	send remote enable	C
tca	take control asynchronously	C
tcs	take control synchronously	AH, C
ton	talk only	T, TE
Local messages sent (to device functions)		
	Non-defined	
	See message output tables in section 2 for description of device function interaction which provides guidelines as to the appropriate states from which local messages may be sent to the device functions.	
Remote messages received		
ATN	ATTENTION	SH, AH, T, TE, L, LE, PP, C
DAB	DATA BYTE	(via L, LE)
DAC	DATA ACCEPTED	SH
DAV	DATA VALID	AH
DCL	DEVICE CLEAR	DC
END	END	(via L, LE)
GET	GROUP EXECUTE TRIGGER	DT
GTL	GO TO LOCAL	RL
IDY	IDENTIFY	PP, L, LE
IFC	INTERFACE CLEAR	T, TE, L, LE, C
LLO	LOCAL LOCKOUT	RL
MLA	MY LISTEN ADDRESS	L, LE, RL
[MLA]	MY LISTEN ADDRESS	T
MSA or [MSA]	MY SECONDARY ADDRESS	TE, LE
MTA	MY TALK ADDRESS	T, TE
[MTA]	MY TALK ADDRESS	L
OSA	OTHER SECONDARY ADDRESS	TE
OTA	OTHER TALK ADDRESS	T, TE
PCG	PRIMARY COMMAND GROUP	TE, LE, PP
PPC	PARALLEL POLL CONFIGURE	PP
[PPD]	PARALLEL POLL DISABLE	PP

(continued on page 111)

Mnemonic	Message	Interface function(s)
Remote messages received *(continued)*		
[PPE]	PARALLEL POLL ENABLE	PP
PPRn	PARALLEL POLL RESPONSE n	(via C)
PPU	PARALLEL POLL UNCONFIGURE	PP
REN	REMOTE ENABLE	RL
RFD	READY FOR DATA	SH
RQS	REQUEST SERVICE	(via L, LE)
[SDC]	SELECTED DEVICE CLEAR	DC
SPD	SERIAL POLL DISABLE	T, TE
SPE	SERIAL POLL ENABLE	T, TE
SRQ	SERVICE REQUEST	(via C)
STB	STATUS BYTE	(via L, LE)
TCT or [TCT]	TAKE CONTROL	C
UNL	UNLISTEN	L, LE
Remote messages sent		
ATN	ATTENTION	C
DAB	DATA BYTE	(via T, TE)
DAC	DATA ACCEPTED	AH
DAV	DATA VALID	SH
DCL	DEVICE CLEAR	(via C)
END	END	(via T)
GET	GROUP EXECUTE TRIGGER	(via C)
GTL	GO TO LOCAL	(via C)
IDY	IDENTIFY	C
IFC	INTERFACE CLEAR	C
LLO	LOCAL LOCKOUT	(via C)
MLA or [MLA]	MY LISTEN ADDRESS	(via C)
MSA or [MSA]	MY SECONDARY ADDRESS	(via C)
MTA or [MTA]	MY TALK ADDRESS	(via C)
OSA	OTHER SECONDARY ADDRESS	(via C)
OTA	OTHER TALK ADDRESS	(via C)
PCG	PRIMARY COMMAND GROUP	(via C)
PPC	PARALLEL POLL CONFIGURE	(via C)
[PPD]	PARALLEL POLL DISABLE	(via C)
[PPE]	PARALLEL POLL ENABLE	(via C)
PPRn	PARALLEL POLL RESPONSE n	PP
PPU	PARALLEL POLL UNCONFIGURE	(via C)
REN	REMOTE ENABLE	C
RFD	READY FOR DATA	AH
RQS	REQUEST SERVICE	T, TE
[SDC]	SELECTED DEVICE CLEAR	(via C)
SPD	SERIAL POLL DISABLE	(via C)
SPE	SERIAL POLL ENABLE	(via C)
SRQ	SERVICE REQUEST	SR
STB	STATUS BYTE	(via T, TE)
TCT	TAKE CONTROL	(via C)
UNL	UNLISTEN	(via C)

Annex E
(informative)

Multiline interface messages – ISO code representation

MULTILINE INTERFACE MESSAGES: ISO 7-BIT CODE REPRESENTATION (ISO 646 international version)
(sent and received with ATN = 1)

b4	b3	b2	b1	Bits 2) b_7 b_6 b_5 → / Row ↓ Column →	0 0 0	MSG 1)	0 0 1	MSG	0 1 0	MSG	0 1 1	MSG	1 0 0	MSG	1 0 1	MSG	1 1 0	MSG	1 1 1	MSG
					0		1		2		3		4		5		6		7	
0	0	0	0	0	NUL		DLE		SP	MLA assigned to device	0	MLA assigned to device	●	MTA assigned to device	P	MTA assigned to device		Meaning defined by PCG code	p	Meaning defined by PCG code
0	0	0	1	1	SOH	GTL	DC1	LLO	!		1		A		Q		a		q	
0	0	1	0	2	STX		DC2		"		2		B		R		b		r	
0	0	1	1	3	ETX		DC3		#		3		C		S		c		s	
0	1	0	0	4	EOT	SDC	DC4	DCL	$		4		D		T		d		t	
0	1	0	1	5	ENQ	PPC 3)	NAK	PPU	%		5		E		U		e		u	
0	1	1	0	6	ACK		SYN		&		6		F		V		f		v	
0	1	1	1	7	BEL		ETB		'		7		G		W		g		w	
1	0	0	0	8	BS	GET	CAN	SPE	(		8		H		X		h		x	
1	0	0	1	9	HT	TCT	EM	SPD	)		9		I		Y		i		y	
1	0	1	0	10	LF		SUB		*		:		J		Z		j		z	
1	0	1	1	11	VT		ESC		+		;		K		[		k		(	
1	1	0	0	12	FF		FS		,		<		L		\		l		:	
1	1	0	1	13	CR		GS		–		=		M		]		m		)	
1	1	1	0	14	SO		RS		.		>		N		ˆ		n		‾	
1	1	1	1	15	SI		US		/		?	UNL	O		—	5)	o		DEL	5)

Addressed command group (ACG)

Universal command group (UCG)

Listen address group (LAG)

Talk address group (TAG)

Primary command group (PCG)

Parallel poll enable (PPE)

Parallel poll disable (PPD)

Secondary command group (SCG)

1) MSG = interface message.
2) b_1 = DIO1 . . . b_7 = DIO7.
3) Requires secondary command.
4) Dense subset (column 2 through 5).
5) See Clause 39.

Annex F
(informative)

Logic circuit implementation

F.1 To assist the designer in the interpretation of the state diagrams, possible circuit implementations are given for situations occurring within the interface functions. It must be understood that the logic diagrams given in this annex do not show the only implementations possible nor do they even represent "recommended" implementations. They are for educational purposes only.

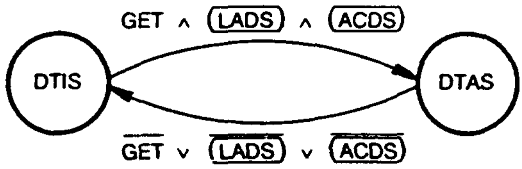
F.2 State diagrams are used to represent two concepts:

- They allow differentiation between the different responses an interface function might produce and identify each with one or more unique states of the interface function.
- They identify those situations where an interface function is required to "remember" past events in order to produce the correct response.

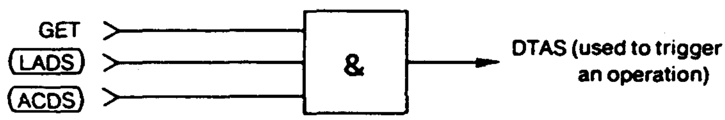
Each state in any of the diagrams serves either or both of these purposes. For example, the LADS state of the Listener interface function has no unique response associated with it and cannot be distinguished from the LIDS state. Its purpose, however, is to "remember" that the device has received a listen address over the bus and is therefore able to enter the LACS state when the ATN message is received false (see figure 7). Conversely, the LACS state is an example of one which has no memory but which exists as a distinct state only to show a specific response capability which does not exist during the LADS state. The only internal difference between the two states is the value of the ATN message and no memory is required since this message value is continuously available.

F.3 Implementing states which require no memory

The Device Trigger interface function is an example of a complete interface function which requires no memory. Its state diagram (from figure 13) is the following:

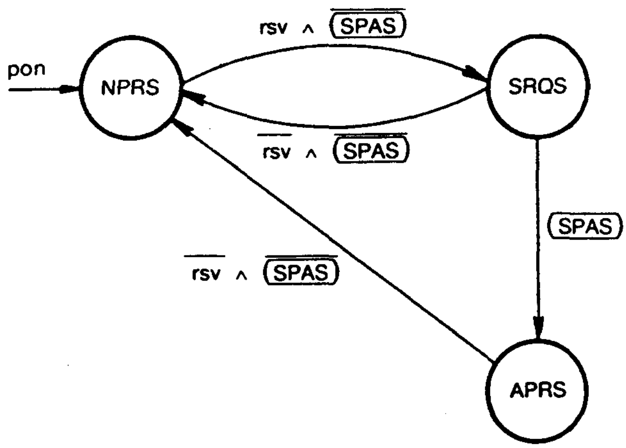


Since no memory is required, this interface function can be implemented with a single AND gate:

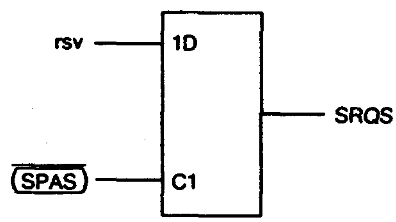


F.4 Implementing states which require memory

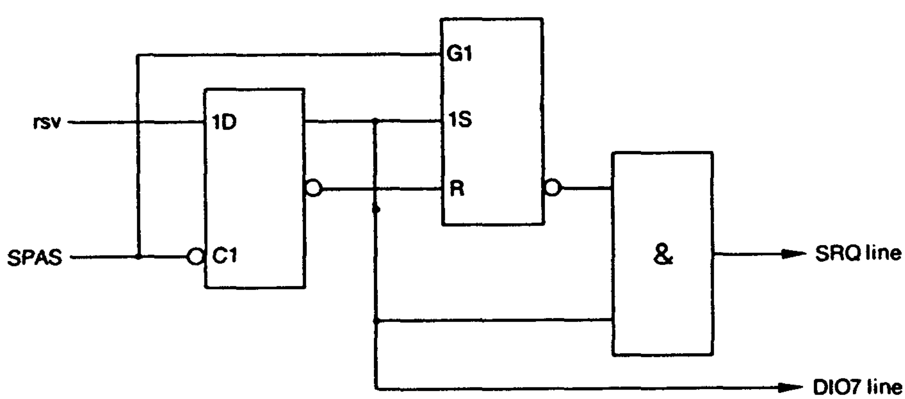
The Service Request interface function is an example requiring memory for its states. Its state diagram (from figure 9) is the following:



The two top states taken by themselves represent a circuit whose internal state follows the value of the *rsv* message but only if (SPAS) is false. This is a standard D-latch:



To complete the circuit all that is needed is a "memory" that (SPAS) has occurred after the latch has turned on. This circuit can be built around a standard RS flip-flop and added to the latch to produce:

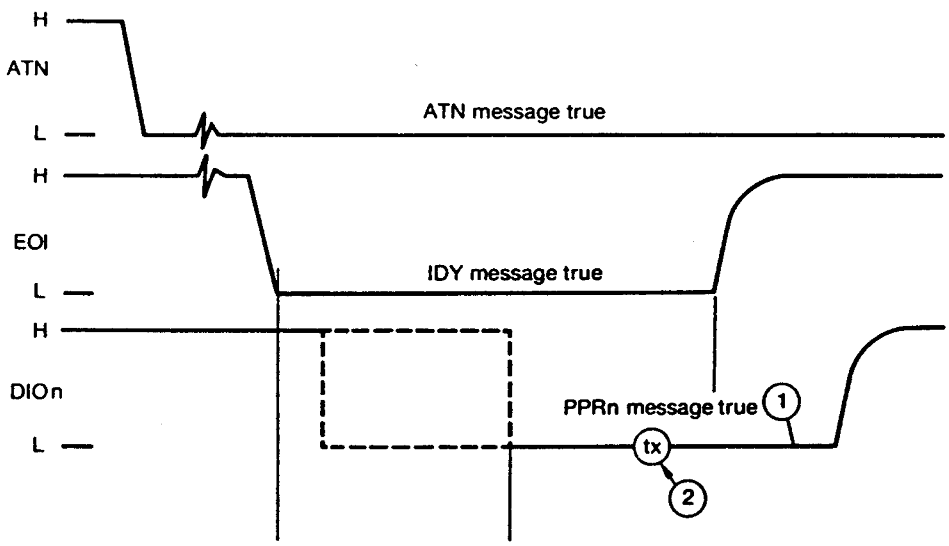


In this circuit, the R/S output stage is forced clear whenever the value of the latched *rsv* message is false. When the latched *rsv* message becomes true, it remains cleared until the first time the SPAS state becomes active at which time it sets, "remembering" that an RQS message has been sent and SRQ no longer needs to be held true.

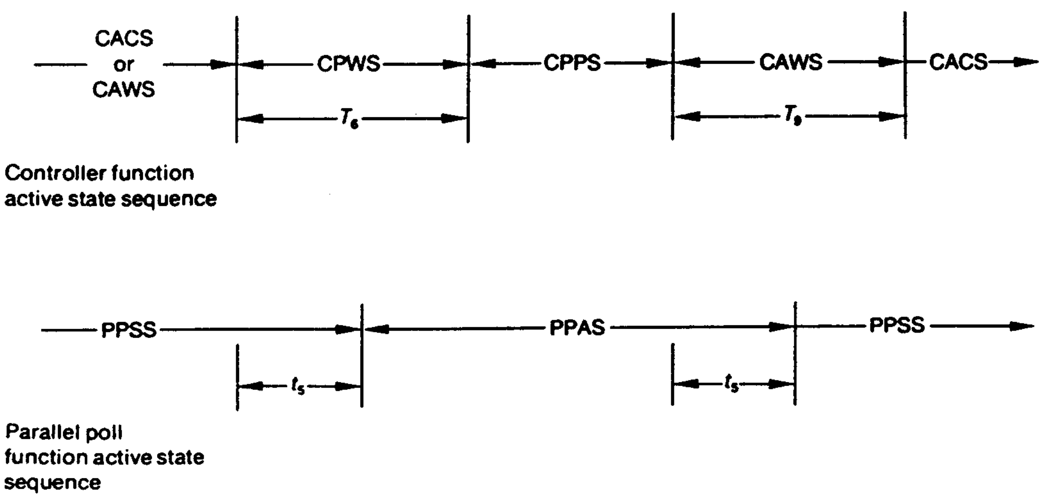
Annex G
(informative)

Parallel poll sequence

G.1 Signal line waveforms



G.2 Interface function active states



NOTES

1 PPRn message true shown in one of two alternate states as determined by the PPE message.

2 Strobe of DIOn lines occurs internal to the controller at any time during the CPPS state by a designer-defined method (the transfer of status data bits during parallel poll does not utilize the handshake process).

Annex H
(informative)

Description of interface parameters on data sheets

This set of guidelines is intended to facilitate the preparation of documentation, particularly data sheets, related to products meeting the requirements of this part. A product meeting the requirements of this part may also have additional device function capability; reference to such capabilities which are accessible via the interface port of this part may be useful to instrumentation systems users.

This annex H forms the basis of a CHECKLIST for both designer and user to describe important interface-related parameters, hence not all parameters listed are expected to be used in every product. The parameters listed are intended as a general guideline and should not be regarded as exhaustive. The amount of material that can be provided on any data sheet depends on space available and the major purpose for which it is intended (e.g. detailed technical data sheet, general descriptive brochure, abbreviated listing in operating manuals).

H.1 General

It is recommended that the data sheet for an instrument or device meeting the requirements of this part contain information which will enable the user of that device to analyse its general capability, programmability, and system performance (relative to the interface). Similarly, it is recommended that the manual supplied with the instrument provide a more detailed description of the interface-related capabilities to facilitate the configuration of instrumentation systems. The descriptive information should take account of this part 1 and also part 2 of IEC 625. Sections 5 and 6 of this part require the device designer to identify, and the device user to be familiar with, the interface capabilities.

It should be noted that for full operational systems it is necessary to have detailed knowledge of the device-dependent characteristics of each device in a system (such detailed knowledge may require information additional to that contained in this part), therefore, compatibility, in the total sense, is not necessarily obtained even though the requirements of this part have been met. In addition to the following recommendations, the designer should include with the user information details about the programmable device-dependent functions of the instrument.

H.2 Description of interface function capabilities

It is recommended that data sheets indicate, in symbolic form as a minimum, the set of interface functions provided by the subject device. Short descriptive phrases as shown, may be useful where space permits. Not all interface functions need be included in a product, in which case "no capability" may be expressed by the function mnemonic and the number 0 (e.g. C0).

H.2.1 *Source Handshake Function (SH)*

SH1 Complete capability.

H.2.2 *Acceptor Handshake function (AH)*

AH1 Complete capability.

H.2.3 *Talker Function (T)*

T1 Basic Talker, Serial Poll, Talk Only Mode.
T2 Basic Talker, Serial Poll.
T3 Basic Talker, Talk Only Mode.
T4 Basic Talker.
T5 Basic Talker, Serial Poll, Talk Only Mode, Unaddress if MLA.
T6 Basic Talker, Serial Poll, Unaddress if MLA.
T7 Basic Talker, Talk Only Mode, Unaddress if MLA.
T8 Basic Talker, Unaddress if MLA.

H.2.4 *Extended Talker Function (TE)*

TE1 Basic Extended Talker, Serial Poll, Talk Only Mode.
TE2 Basic Extended Talker, Serial Poll.
TE3 Basic Extended Talker, Talk Only Mode.
TE4 Basic Extended Talker,
TE5 Basic Extended Talker, Serial Poll, Talk Only Mode, Unaddress if MSA ^ LPAS.
TE6 Basic Extended Talker, Serial Poll, Unaddress if MSA ^ LPAS.
TE7 Basic Extended Talker, Talk Only Mode, Unaddress if MSA ^ LPAS.
TE8 Basic Extended Talker, Unaddress if MSA ^ LPAS.

H.2.5 *Listener Function (L)*

L1 Basic Listener, Listen Only Mode.
L2 Basic Listener.
L3 Basic Listener, Listen Only Mode, Unaddress if MTA.
L4 Basic Listener, Unaddress if MTA.

H.2.6 *Extended Listener Function (LE)*

LE1 Basic Extended Listener, Listen Only Mode.
LE2 Basic Extended Listener.
LE3 Basic Extended Listener, Listen Only Mode, Unaddress if MSA ^ TPAS.
LE4 Basic Extended Listener, Unaddress if MSA ^ TPAS.

H.2.7 *Service Request Function (SR)*

SR1 Complete capability.

H.2.8 *Remote Local Function (RL)*

RL1 Complete capability.
RL2 Without local lock out.

H.2.9 *Parallel Poll Function (PP)*

PP1 Remote Configuration.
PP2 Local Configuration only.

H.2.10 *Device Clear Function (DC)*

DC1 Complete capability.
DC2 Without selective device clear.

H.2.11 *Device Trigger Function (DT)*

DT1 Complete capability.

H.2.12 *Controller Function (C)*

Symbol	*Short description*
C1	System Controller
C2	Send IFC and Take Charge
C3	Send REN
C4	Respond to SRQ

NOTE - Only one of the following applies to a given product.

C5	Send Interface Messages, Receive Control, Pass Control, Pass Control to Self, Parallel Poll, Take Control, Synchronously.
C6	Send Interface Messages, Receive Control, Pass Control, Pass control to Self, Parallel Poll.
C7	Send Interface Messages, Receive Control, Pass Control, Pass Control to Self,Take Control Synchronously.
C8	Send Interface Messages, Receive Control, Pass Control, Pass Control to Self.
C9	Send Interface Messages, Receive Control, Pass Control, Parallel Poll, Take Control Synchronously.
C10	Send Interface Messages, Receive Control, Pass Control, Parallel Poll.
C11	Send Interface Messages, Receive Control, Pass Control, Take Control Synchronously.
C12	Send Interface Messages, Receive Control, Pass Control.
C13	Send Interface Messages, Receive Control, Parallel Poll, Take Control Synchronously.
C14	Send Interface Messages, Receive Control, Parallel Poll.
C15	Send Interface Messages, Receive Control, Take Control Synchronously.
C16	Send Interface Messages, Receive Control.
C17	Send Interface Messages, Pass Control, Pass Control to Self, Parallel Poll, Take Control Synchronously.
C18	Send Interface Messages, Pass Control, Pass Control to Self, Parallel Poll.
C19	Send Interface Messages, Pass Control, Pass Control to Self, Take Control Synchronously.
C20	Send Interface Messages, Pass Control, Pass Control to Self.

C21	Send Interface Messages, Pass Control, Parallel Poll, Take Control Synchronously.
C22	Send Interface Messages, Pass Control, Parallel Poll.
C23	Send Interface Messages, Pass Control, Take Control Synchronously.
C24	Send Interface Messages, Pass Control.
C25	Send Interface Messages, Parallel Poll, Take Control Synchronously.
C26	Send Interface Messages, Parallel Poll.
C27	Send Interface Messages, Take Control Synchronously.
C28	Send Interface Messages.

H.3 Description of typical time-related values

The description of time-related values in this section is highly dependent upon the total system configuration (i.e. the nature of the talker, listener, and controller devices). Actual time values may be highly dependent upon the actual measurement conditions, the device-dependent nature of the instrument being specified, and possibly the operating system software located in either instrument or controller. Therefore, it is difficult and possibly irrelevant to specify precise values.

H.3.1 *Data rates for DAB messages*

a) Data input rate (when addressed to listen)

For example: N kilobytes per second
and relevant conditions (e.g. configuration,
data type, functional operation)

b) Data output rate (when addressed to talk)
For example: N kilobytes per second
and relevant conditions (e.g. configuration,
data type, functional operation)

H.3.2 *Other time values*

EXAMPLES

a) Interface handshake delays (e.g. time out, hold)

b) Response time to device commands

c) Response time to interface messages

H.4 Additional information of value to systems users and designers

H.4.1 *Functional specifications*

Any operating modes that deviate from this part should be explicitly stated and explained.

The standard (IEC 625) to which the device conforms should be listed with its full title with information on obtaining such standards.

The means of generating and using the R/L function and the *rtl* local message should be documented.

The response to Group Execute Trigger should be documented.

H.4.2 *Controls, connectors and indicators*

All IEC 625-related switches should be illustrated or their locations described. In addition, the following information should be documented:

How to set primary and secondary (if any) addresses.

How to read addresses from the front panel (if possible).

Effects of other IEC 625 alterable modes (such as *ton*, *lon*, etc.).

H.4.3 *Power up/down sequences and default values*

Describe *pon* self-test functionality and abnormal condition reporting.

Describe the effect of power-interrupt.

Describe non-volatile memory features.

Describe default device state at *pon.*

H.4.4 *Programmable device functions*

List which device functions are bus controllable.

Describe I/O buffering (if any), for example:

- buffer sizes;
- maximum lines accepted;
- maximum number of digits.

List limits of numerical parameters (if any), for example:

- mantissa and exponent limits;
- internal precision for rounding and rounding rules;
- limits on parameters and effect of going beyond.

H.4.5 *Status handling information*

Identify if any *rsv* local messages may be inhibited.

List all conditions that may cause the device to set *rsv* TRUE.

H.5 Electrical driver/receiver capabilities

H.5.1 *Signal lines with open collector drivers = E1*

(Specify applicable DIO lines and constraints)

H.5.2 *Signal lines with three-state drivers = E2*

(Specify applicable DIO lines and constraints)

Annex J
(informative)

Recommended methods for improving the screening of cables that are specified in this standard

This part specifies a connector and cable assembly that provides acceptable EMC performance under many circumstances. In some applications, however, it is either advisable or required to further reduce the effects of EM radiation. The recommended method provides general guidance on precautions that can be taken to minimize the effects of EM radiation on instrumentation systems to levels below those which would otherwise be obtained by complying with the specifications given elsewhere in this part.

It is generally accepted that the cable assembly (i.e. the interconnecting cable and the associated connectors) and the connector/cable interface immediately inside the instrument will, to a large extent, determine the EMC performance of the overall instrumentation system.

The following guidelines describe how to connect instruments fitted with connectors specified in this standard to obtain improved EMC performance. It should be noted such screening can be provided to reduce the effects of four basic types of interference:

a) Leakage from cables and connectors which then affect other RF systems.

b) External interference affecting cables and connectors, and consequently data transfer as decribed in this part.

c) Injection of signals between any signal ground return, logic ground and shield.

d) External interference which may be injected between any signal ground return, logic ground and shield.

J.1 Reducing the effect of radiated interference

These goals can be accomplished by using cabling techniques with minimum leakage.

The basic recommendations detailed below and illustrated in figure J.1 may be of assistance in meeting the screening requirements applicable to instrumentation systems which might be imposed in some countries by their National Regulations.

J.1.1 *Screened cables*

A minimum coverage of 85 % is essential but a 90 % coverage would provide a further improvement. A combination of both braid and metallized mylar ®[1] or foil provides still further improvement in cable screening and is recommended.

[1] Mylar ® is the trade name of a product supplied by Du Pont de Nemours SA. This information is given for the convenience of users of this International Standard and does not constitute an endorsement by the IEC of the product named. Equivalent products may be used if they can be shown to lead to the same results.

J.1.2 *RF connection techniques*

It is insufficient to rely only on ground connection between the cable screen and the connector on the equipment through one or even several connector pins. Instead, an adequate RF connection to chassis, for example, through the metallic connector housing is required. Care should be observed to provide backward compatibility (mechanical and electrical) with the already existing shapes of connector housing.

J.1.3 *Connector housing*

The connector housing should be metallic. Since the housing is to be used for chassis connection the parts of the housing which provide the electrical contact should be designed and finished accordingly (e.g. metallic and conductive housing, metal shelled connectors or equivalent, good RF connection between cable screen and metallic housing, etc.).

J.1.4 *EM radiation screening*

EM radiation screening can be further improved by providing a metallic cover over the ends of the "piggyback" connectors. This measure will, however, only be necessary in exceptional cases. The contribution to RF leakage by the connector contacts is small.

J. 2 Connectors conforming to this part used on equipment

J.2.1 The IEC 625 connector port should have a metal shell and ensure adequate RF contact with the cable connector housing (in accordance with J.1.3).

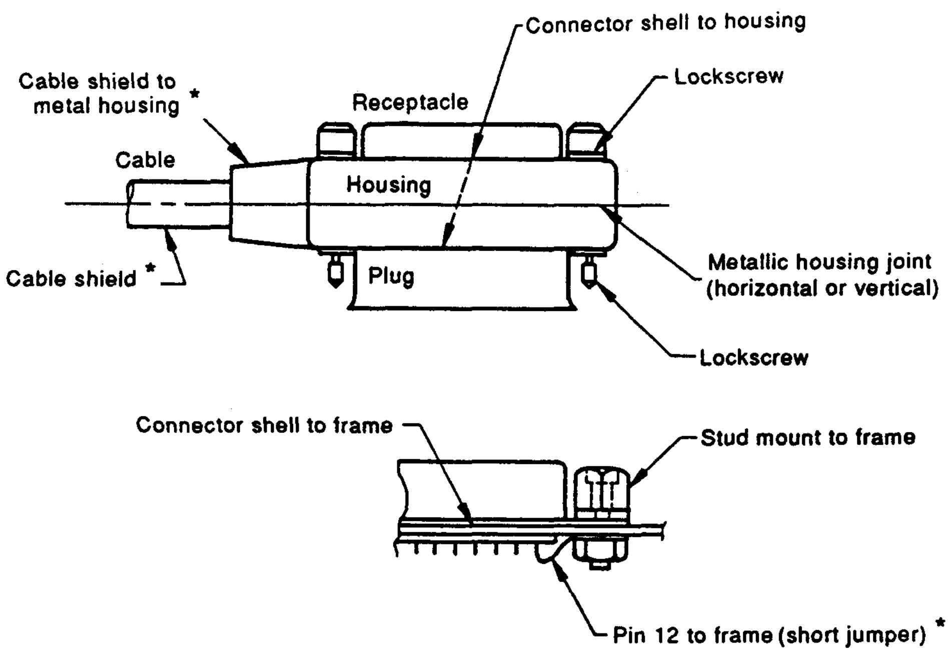
J.2.2 The electrical connection between the "shield" pin of the connector, the metal housing and shell of the chassis mounted connector, and chassis frame should be made via as short a route as possible. Noticeable performance degradation may occur with ground lengths in excess of about 50 mm. Therefore, when using a flat cable to make the internal connections to the IEC 625 connector port, the screen should not be connected to chassis via this cable. This does not apply to the logic signal grounds.These may be connected along the same path as the signal lines.

J.3 Reducing the effects of conducted interference

Signals which exist on the signal ground lines can cause conducted electromagnetic interference inside a device when ground loops are formed as the signal ground lines are connected together inside the device.

Such ground loops can also create Electromagnetic Interference, EMI, as signals are generated by the effect of RF fields on such ground loops.

It is recommended that a device minimize its susceptibility to and generation of conducted interference among the logic ground and earth conductors (i.e. connector pins 12, 18, 19, 20, 21, 22, 23, 24) by using low-impedance RF ground returns as close to the connector port as possible within the device.



NOTES

1 All surfaces pointed to are metallic.

2 Good quality electrical contact at all points recommended.

3 Most critical conductive points (*) where good RF practice is essential.

Figure J.1 – RF contact (ground) points

Annex K
(informative)

Information on the 25-pin connector

Introduction

It is current practice for the 24-pin connector, now specified in section 4, to be used instead of the 25-pin connector which section 4 previously specified. It is recommended that this 24-pin connector is used in new device designs and implementations.

This annex K provides information on the 25-pin connector originally contained in section 4. It is given mainly as a reference for older designs of devices using the 25-pin connector. An interconnection table for use in systems where it is required to use a mixture of devices using 24-pin and 25-pin connectors is given in K.5.

Its application is in interface systems to be used in environments where:

- physical distances between devices are short;
- star or linear bus interconnection networks are useful;
- connector mounting space is limited.

K.1 Connector type

The type of connector intended to be used with this interface is similar to the 25-pin connector specified in IEC 807-2.

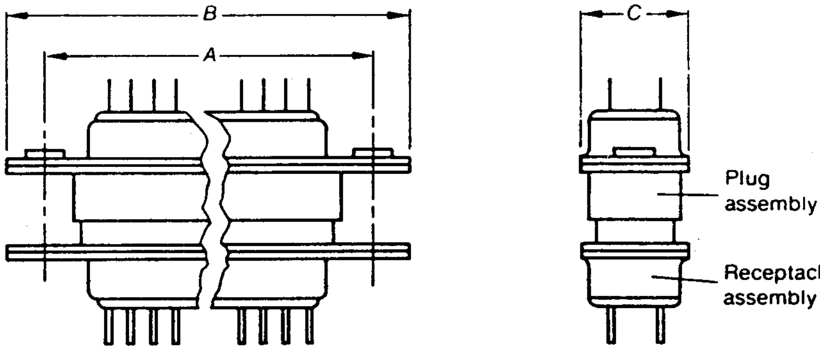
K.1.1 *Electrical requirements*

Voltage rating:	60 V
Test voltage:	0,5 kV (according to IEC 348)
Current rating:	5 A per contact
Contact resistance:	less than 20 mΩ
Insulation resistance:	higher than 1 GΩ

K.1.2 *Mechanical requirements*

Number of contacts:	25
Shell material:	Corrosion resistant plating
Contact material:	gold-plated alloy
Endurance:	more than 1 000 insertions to reach 20 mΩ contact resistance
Wire diameter:	termination to accept a wire of at least 0,35 mm^2

Typical dimensions:



A = 47,17 mm max., 46,91 mm min.
B = 53,42 mm max., 52,65 mm min.
C = 12,93 mm max., 12,17 mm min.

K.1.3 *Environmental requirements*

Basic environmental performance relative to the cold, dry heat and damp-heat tests for climatic category 25/070/21 as explained in annex A of IEC 68-1.

K.2 Contact assignments

The contact assignment of the cable connector and the device connector shall be as shown below:

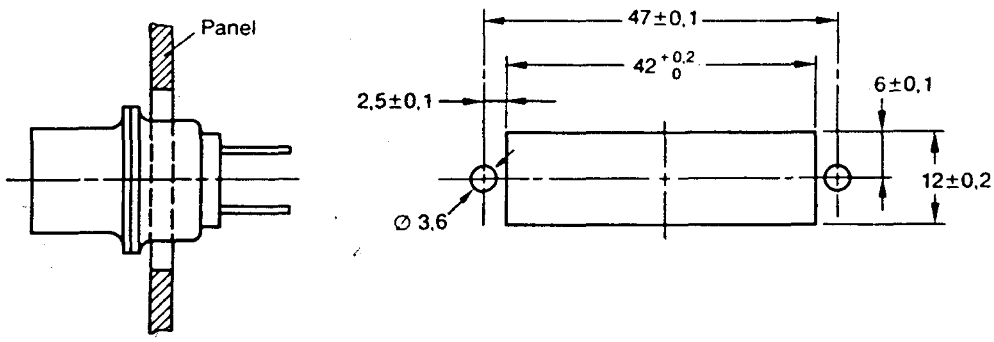
Contact	Signal line	Contact	Signal line
1	DIO 1	14	DIO 5
2	DIO 2	15	DIO 6
3	DIO 3	16	DIO 7
4	DIO 4	17	DIO 8
5	REN	18	Gnd (5)
6	EOI	19	Gnd (6)
7	DAV	20	Gnd (7)
8	NRFD	21	Gnd (8)
9	NDAC	22	Gnd (9)
10	IFC	23	Gnd (10)
11	SRQ	24	Gnd (11)
12	ATN	25	Gnd (12)
13	Shield		

NOTE - "Gnd (*n*)" refers to the suggested signal ground return of the reference contact *n*.

K.3 Device connector mounting

K.3.1 Each device shall be provided with a plug-type connector. The connector mounting shall make provisions to accept the locking screws of the cable assembly.

K.3.2 The connector shall be mounted on the device in accordance with the mechanical dimensions of figure K.1.



NOTE - The connector may also be mounted inside the panel with proper attention to critical dimensions.

Figure K.1 – Connector mounting and panel cutout

K.3.3 The mounting positions of the connector on a device, as viewed from the rear of the device in its normal operating position are shown in figure K.2. The horizontal position is preferred. The connector location should allow a minimum bend radius of 40 mm for cable clearance.

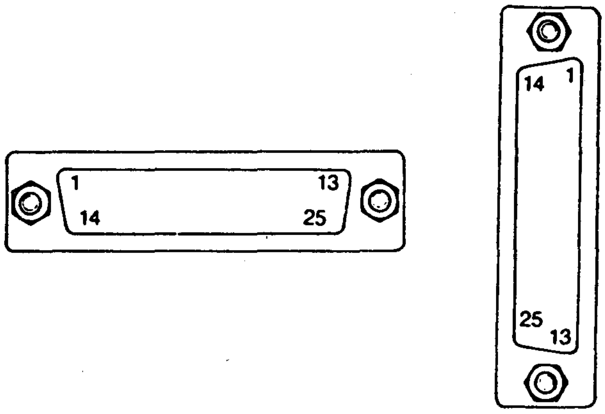
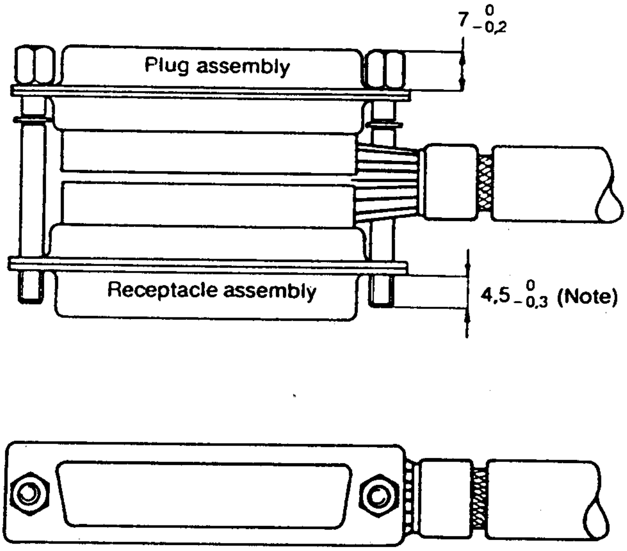


Figure K.2 – Mounting positions

K.4 Cable assembly

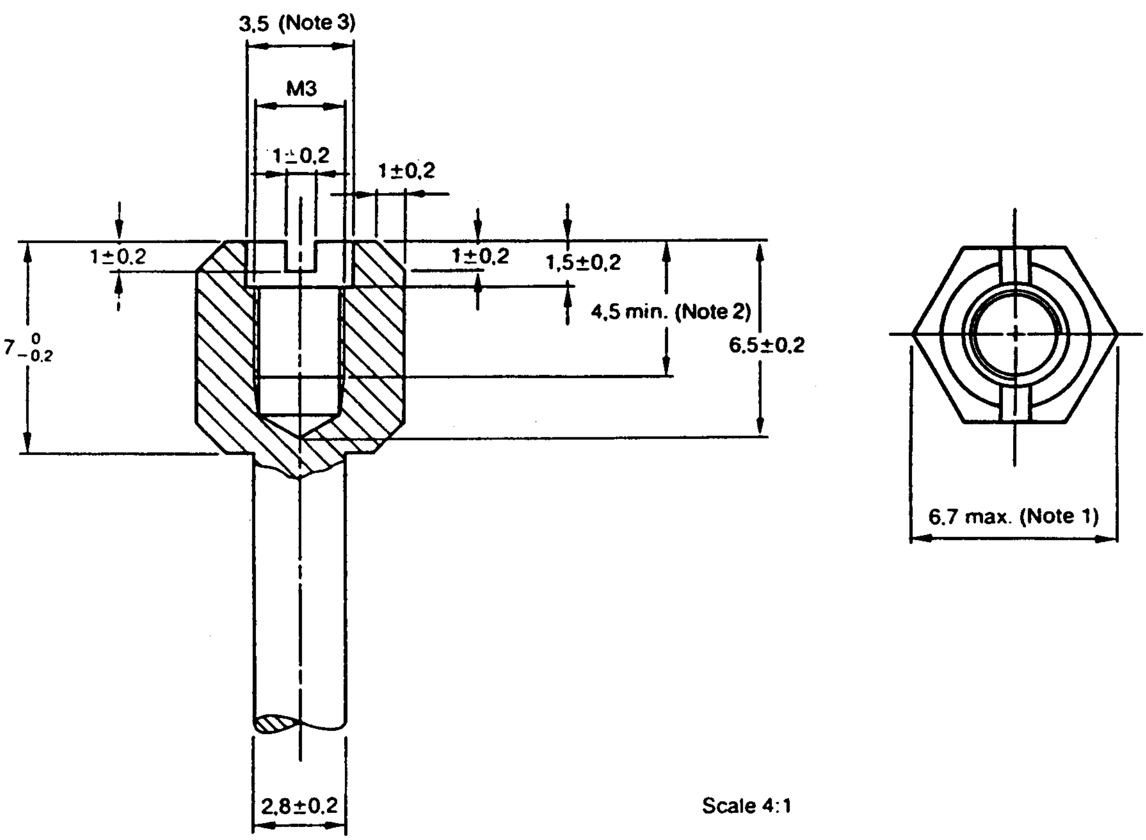
K.4.1 The cable assembly shall be provided with both a plug and a receptacle connector type at each end of the cable so that one connector can be stacked on top of another in piggyback fashion as shown in figure K.3.



NOTE - Length of thread M3.

Figure K.3 – Cable connector assembly

K.4.2 Each connector assembly shall be fitted with two captive locking screws in accordance with figure K.3 above and figure K.4.



NOTES

1 Head may be hexagonal or round.

2 Depth of thread M3.

3 Counterbore and slot for screwdriver optional.

Figure K.4 – Locking screw

K.4.3 It is recommended that each pair of connectors, assembled in accordance with K.4.1 be partially enclosed within a suitable housing as shown in figure K.5.

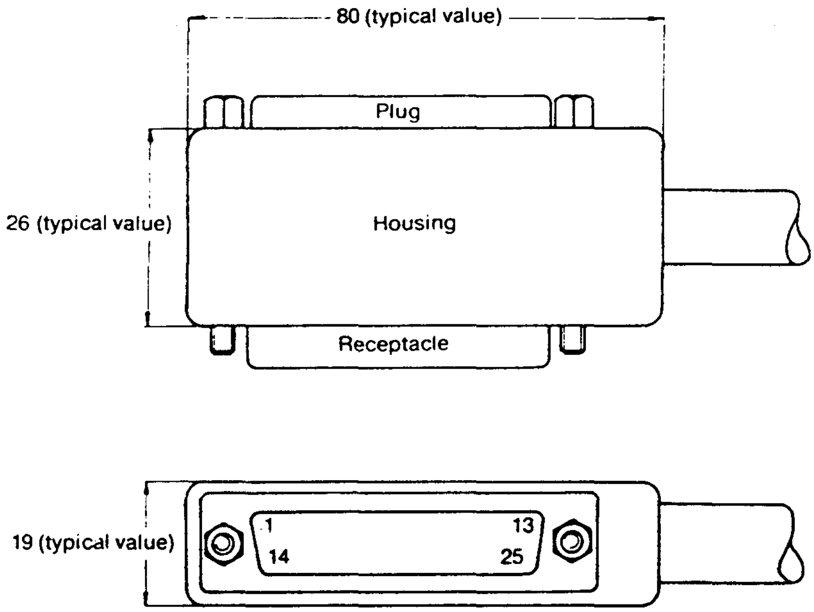


Figure K.5 – Cable connector housing

K.5 Interconnection table

Should it be necessary to interconnect devices having a mixture of both 24- and 25-pin connector assemblies, the following wiring should be used in the adapter cable assembly.

24-Pin connector	Signal	25-Pin connector
1	DIO 1	1
2	DIO 2	2
3	DIO 3	3
4	DIO 4	4
5	EOI	6
6	DAV	7
7	NRFD	8
8	NDAC	9
9	IFC	10
10	SRQ	11
11	ATN	12
12	SHIELD	13
13	DIO 5	14
14	DIO 6	15
15	DIO 7	16
16	DIO 8	17
17	REN	5
18	GND	20
19	GND	21
20	GND	22
21	GND	23
22	GND	24
23	GND	25
24	GND	18, 19

Annex L
(informative)

Address switch labelling and interface status indicators

L.1 General comments

To assist a device designer, a recommended positioning and labelling for the Talk and Listen address switch is given. A recommendation is also given for the labelling of interface status indicators or the interface status messages to be used on displays.

L.2 Talk and Listen addresses

A device should have a non-volatile local means of pre-selecting the values assigned to its MTA (my Talk address) and its MLA (my Listen address). The value for bits T1 to T5 assigned to MTA or bits L1 to L5 assigned to MLA should be capable of being altered by the device operator. If the MTA and MLA are identical (bits 1 to 5, as in 39.1 and 39.2), then a single common storage element may be used. This discussion will use this single address value but does not preclude the use of separate addresses for MTA and MLA.

L.3 DIP switch

When the common MTA and MLA (bits 1-5) address value is selected by a DIP switch, the DIP switch needs 5 bits (combination of T1 to T5 and L1 to L5) or switch poles to set the address value. The five switch poles should be physically adjacent to each other as indicated in figure L.1 or figure L.2. If the local messages *ton* and/or *lon* are implemented as poles on the same DIP switch, they should be placed as shown in figure L.1 or figure L.2. In the case that *ton* is not implemented but *lon* is, the *lon* switch should be immediately to the left or above the address switches.

Where feasible, the DIP switch should be externally accessible and labelled similar to figure L.1 or figure L.2. The description of the switch and its location should be clearly visible in the device's operation manual.

Labelling associated with the DIP switch is to include the binary weighting applied to each DIP switch position to facilitate determination of the address value. An illustration of this is shown in figure L.1 and figure L.2. An indication of which switches are used to set the address should also be provided. Diagrams in the instruction manual should be provided to clearly indicate which physical position of the individual DIP switches provides an "0/1" condition. The "1" position for the switches is to present a logical TRUE state on the IEC 625 interface bus while the "0" position is to present a logical FALSE state on the bus. The labelled "X" positions show a sample selection of the switches.

```
/------------------------------------------------+
  +-+    +-+    +-+    +-+    +-+    +-+    +-+   |
  | |    |X|    | |    |X|    | |    |X|    |X|   |
  |-|    |-|    |-|    |-|    |-|    |-|    |-|   |
  |X|    | |    |X|    | |    |X|    | |    | |   | ton,
  +-+    +-+    +-+    +-+    +-+    +-+    +-+   | address (01011)
/------------------------------------------------+
  lon    ton  | 16     8      4      2      1     |
              |           Address               |
              |_________________________________|
```

Figure L.1 – Labelling diagram for DIP switch in horizontal orientation

```
/                 /
|  +,X,+---+      |  lon
|  +'X'+---+      |
|                 |
|  +---+,X,+      |  ton
|  +---+'X'+      |
|                 |--------
|  +---+,X,+      |        |
|  +---+'X'+      |  16    |
|                 |        |
|  +---+,X,+      |        |
|  +---+'X'+      |   8 A  |
|                 |     d  |
|  +,X,+---+      |     d  |  ton, address (11010)
|  +'X'+---+      |   4 r  |
|                 |     e  |
|  +---+,X,+      |     s  |
|  +---+'X'+      |   2 s  |
|                 |        |
|  +,X,+---+      |        |
|  +'X'+---+      |   1    |
|                 |--------
+-----------------+
    0      1
```

Figure L.2 – Labelling diagram for DIP switch in vertical orientation

L.4 Alternate implementations

Other methods besides DIP Switches can be used to set the non-volatile value of the MTA and MLA address or change the state of *ton* and *lon* local messages. This alternative method, however, should provide the ability of displaying and modifying the address values. When *ton* and *lon* are implemented, this alternative method should also allow their state to be examined and modified.

L.5 Device status indicators

A device may optionally contain indicators and/or displays showing the current state (or operating mode) of the device interface. When these interface indicators or displays are provided, they should be labelled with or use the abbreviation or name given in the following table. The NDAC, NRFD, or SRQ indicator shall represent the device's view of its remote message and not necessarily the current state of the corresponding interface bus line. An ADDR indicator may be used in place of the TALK and LSTN indicators.

Table L.1 – Device status indicators

Abreviation	Name	Current state
ADDR	Addressed to Talk or Listen	Device is in TADS, TACS, LADS, or LACS
LOCK	Front panel lockout	Device is in RWLS
LSTN	Addressed to Listen	Device is in LADS or LACS
MA	My Address	Current address value
NDAC	Not data accepted	NDAC asserted
NRFD	Not ready for data	NRFD asserted
REM	Remote	Device is in REMS or RWLS
SRQ	Service request	SRQ asserted
TALK	Addressed to Talk	Device is in TADS or TACS
CACT	Controller active	Device is in CTRS, CACS, CSBS, CSHS, CSWS, CAWS, CPWS or CPPS

Bureau of Indian Standards

BIS is a statutory institution established under the *Bureau of Indian Standards Act,* 1986 to promote harmonious development of the activities of standardization, marking and quality certification of goods and attending to connected matters in the country.

Copyright

Review of Indian Standards

Amendments are issued to standards as the need arises on the basis of comments. Standards are also reviewed periodically; a standard along with amendments is reaffirmed when such review indicates that no changes are needed; if the review indicates that changes are needed, it is taken up for revision. Users of Indian Standards should ascertain that they are in possession of the latest amendments or edition by referring to the latest issue of 'BIS Catalogue' and 'Standards : Monthly Additions'.

This Indian Standard has been developed from Doc : No. LTD 21 (1853).

Amendments Issued Since Publication

Amend No.	Date of Issue	Text Affected

BUREAU OF INDIAN STANDARDS

Headquarters:

Manak Bhavan, 9 Bahadur Shah Zafar Marg, New Delhi 110 002
Telephones: 323 01 31, 323 33 75, 323 94 02

Telegrams: Manaksanstha
(Common to all offices)

Regional Offices:	Telephone
Central: Manak Bhavan, 9 Bahadur Shah Zafar Marg NEW DELHI 110002	323 76 17, 323 38 41
Eastern: 1/14 C. I. T. Scheme VII M, V. I. P. Road, Kankurgachi KOLKATA 700 054	337 84 99, 337 85 61, 337 86 26, 337 91 20
Northern: SCO 335-336, Sector 34-A, CHANDIGARH 160 022	60 38 43, 60 20 25
Southern: C. I. T. Campus, IV Cross Road, CHENNAI 600 113	254 12 16, 254 14 42, 254 25 19, 254 13 15
Western: Manakalaya, E9 MIDC, Marol, Andheri (East) MUMBAI 400 093	832 92 95, 832 78 58, 832 78 91, 832 78 92

Branches: AHMADABAD. BANGALORE. BHOPAL. BHUBANESHWAR. COIMBATORE. FARIDABAD. GHAZIABAD. GUWAHATI. HYDERABAD. JAIPUR. KANPUR. LUCKNOW. NAGPUR. NALAGARH. PATNA. PUNE. RAJKOT. THIRUVANANTHAPURAM.

Printed at New India Printing Press, Khurja, India